# Unsupervised Multi-Domain Adaptation with Feature Embeddings

**Yi Yang** and **Jacob Eisenstein**
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30308
{yiyang+jacobe}@gatech.edu

## Abstract

Representation learning is the dominant technique for unsupervised domain adaptation, but existing approaches have two major weaknesses. First, they often require the specification of "pivot features" that generalize across domains, which are selected by task-specific heuristics. We show that a novel but simple *feature embedding* approach provides better performance, by exploiting the feature template structure common in NLP problems. Second, unsupervised domain adaptation is typically treated as a task of moving from a single source to a single target domain. In reality, test data may be diverse, relating to the training data in some ways but not others. We propose an alternative formulation, in which each instance has a vector of *domain attributes*, can be used to learn distill the domain-invariant properties of each feature.[1]

## 1 Introduction

Domain adaptation is crucial if natural language processing is to be successfully employed in high-impact application areas such as social media, patient medical records, and historical texts. Unsupervised domain adaptation is particularly appealing, since it requires no labeled data in the target domain. Some of the most successful approaches to unsupervised domain adaptation are based on representation learning: transforming sparse high-dimensional surface features into dense vector representations, which are often more robust to domain shift (Blitzer et al., 2006; Glorot et al., 2011). However, these methods are computationally expensive to train, and often require special task-specific heuristics to select good "pivot features."

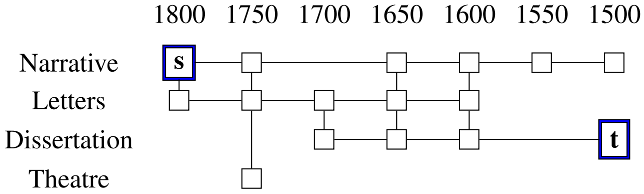
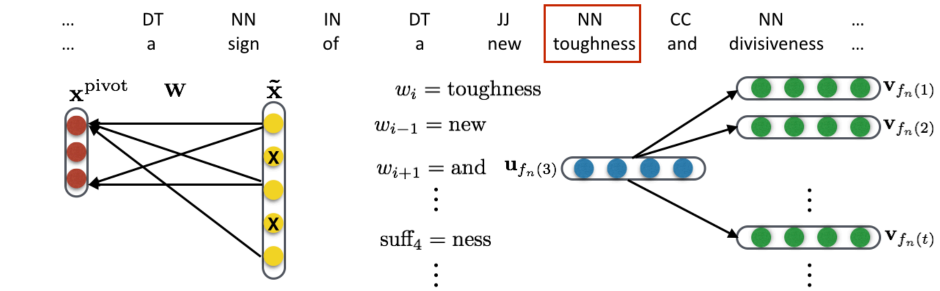


Figure 1: Domain graph for the Tycho Brahe corpus (Galves and Faria, 2010). **Suppose we want to adapt from 19th Century narratives to 16th Century dissertations: can unlabeled data from other domains help?**

A second, more subtle challenge for unsupervised domain adaptation is that it is normally framed as adapting from a single source domain to a single target domain. For example, we may be given part-of-speech labeled text from 19th Century narratives, and we hope to adapt the tagger to work on academic dissertations from the 16th Century. This ignores text from the intervening centuries, as well as text that is related by genre, such as 16th Century narratives and 19th Century dissertations (see Figure 1). We address a new challenge of *unsupervised multi-domain adaptation*, where the goal is to leverage this additional unlabeled data to improve performance in the target domain.[2]

[1] Source code and a demo are available at `https://github.com/yiyang-gt/feat2vec`

[2] Multiple domains have been considered in **supervised** domain adaptation (e.g., Mansour et al., 2009), but these approaches are not directly applicable when there is no labeled data outside the source domain.



(a) Learning a dense transformation $\mathbf{W}$ by reconstructing pivot features.

(b) Directly learning of feature embeddings by maximizing Skip-Gram objective.

Figure 2: Representation learning techniques in structured feature spaces

We present FEMA (**F**eature **EM**beddings for domain **A**daptation), a novel representation learning approach for domain adaptation in structured feature spaces. Like prior work in representation learning, FEMA learns dense features that are more robust to domain shift. However, rather than performing representation learning by reconstructing pivot features, FEMA uses techniques from neural language models to obtain low-dimensional embeddings directly. FEMA outperforms prior work on adapting POS tagging from the Penn Treebank to web text, and it easily generalizes to unsupervised multi-domain adaptation, further improving performance by learning generalizable models across multiple domains.

## 2 Learning feature embeddings

Feature co-occurrence statistics are the primary source of information driving many unsupervised methods for domain adaptation; they enable the induction of representations that are more similar across the source and target domain, reducing the error introduced by domain shift (Ben-David et al., 2010). For example, both Structural Correspondence Learning (SCL; Blitzer et al., 2006) and Denoising Autoencoders (Chen et al., 2012) learn to reconstruct a subset of "pivot features", as shown in Figure 2(a). The reconstruction function — which is learned from unlabeled data in both domains — is then employed to project each instance into a dense representation, which will hopefully be better suited to cross-domain generalization. The pivot features are chosen to be both predictive of the label and general across domains. Meeting these two criteria requires task-specific heuristics; for example, different pivot selection techniques are employed in SCL for syntactic tagging (Blitzer et al., 2006) and sentiment analysis (Blitzer et al., 2007). Furthermore, the pivot features correspond to a small subspace of the feature co-occurrence matrix. In Denoising Autoencoders, each pivot feature corresponds to a dense feature in the transformed representation, but large dense feature vectors impose substantial computational costs at learning time. In SCL, each pivot feature introduces a new classification problem, which makes computation of the cross-domain representation expensive. In either case, we face a tradeoff between the amount of feature co-occurrence information that we can use, and the computational complexity for representation learning and downstream training.

This tradeoff can be avoided by inducing low dimensional feature embeddings directly. We exploit the tendency of many NLP tasks to divide features into *templates*, with exactly one active feature per template (Smith, 2011); this is shown in the center of Figure 2. Rather than treating each instance as an undifferentiated bag-of-features, we use this template structure to induce *feature embeddings*, which are dense representations of individual features. Each embedding is selected to help predict the features that fill out the other templates: for example, an embedding for the current word feature is selected to help predict the previous word feature and successor word feature, and vice versa; see Figure 2(b). The embeddings for each active feature are then concatenated together across templates, giving a dense representation for the entire instance.

Our approach is motivated by word embeddings, in which dense representations are learned for individual words based on their neighbors (Turian et al., 2010; Xiao and Guo, 2013), but rather than learning a single embedding for each word, we learn embeddings for each *feature*. This means that the embedding of, say, 'toughness' will differ depending on whether it appears in the current-word template or the previous-word template (see Table 6). This provides additional flexibility for the downstream learning algorithm, and the increase in the dimensionality of the overall dense representation can be offset by learning shorter embeddings for each feature. In Section 4, we show that feature embeddings convincingly outperform word embeddings on two part-of-speech tagging tasks.

Our feature embeddings are based on the skip-gram model, trained with negative sampling (Mikolov et al., 2013a), which is a simple yet efficient method for learning word embeddings. Rather than predicting adjacent words, the training objective in our case is to find feature embeddings that are useful for predicting other active features in the instance. For the instance $n \in \{1 \ldots N\}$ and feature template $t \in \{1 \ldots T\}$, we denote $f_n(t)$ as the index of the active feature; for example, in the instance shown in Figure 2, $f_n(t) =$ 'new' when $t$ indicates the previous-word template. The skip-gram approach induces distinct "input" and "output" embeddings for each feature, written $\mathbf{u}_{f_n(t)}$ and $\mathbf{v}_{f_n(t)}$, respectively. The role of these embeddings can be seen in the negative sampling objective,

$$\begin{aligned}\ell_n =& \frac{1}{T}\sum_{t=1}^{T}\sum_{t'\neq t}^{T}\Big[\log\sigma(\mathbf{u}_{f_n(t)}^\top \mathbf{v}_{f_n(t')}) \\ &+ k\mathbb{E}_{i\sim P_{t'}^{(n)}}\log\sigma(-\mathbf{u}_{f_n(t)}^\top \mathbf{v}_i)\Big], \qquad (1)\end{aligned}$$

where $t$ and $t'$ are feature templates, $k$ is the number of negative samples, $P_{t'}^{(n)}$ is a *noise distribution* for template $t'$, and $\sigma$ is the sigmoid function. This objective is derived from noise-contrastive estimation (Gutmann and Hyvärinen, 2012), and is chosen to maximize the unnormalized log-likelihood of the observed feature co-occurrence pairs, while minimizing the unnormalized log-likelihood of "negative" samples, drawn from the noise distribution.

Feature embeddings can be applied to domain adaptation by learning embeddings of all features on the union of the source and target data sets; we consider the extension to multiple domains in the next section. The dense feature vector for each instance is obtained by concatenating the feature embeddings for each template. Finally, since it has been shown that nonlinearity is important for generating robust representations (Bengio et al., 2013), we follow Chen et al. (2012) and apply the hyperbolic tangent function to the embeddings. The augmented representation $\mathbf{x}_n^{(\text{aug})}$ of instance $n$ is the concatenation of the original feature vector and the feature embeddings,

$$\mathbf{x}_n^{(\text{aug})} = \mathbf{x}_n \oplus \tanh[\mathbf{u}_{f_n(1)} \oplus \cdots \oplus \mathbf{u}_{f_n(T)}],$$

where $\oplus$ is vector concatenation.

## 3 Feature embeddings across domains

We now describe how to extend the feature embedding idea beyond a single source and target domain, to unsupervised multi-attribute domain adaptation (Joshi et al., 2013). In this setting, each instance is associated with $M$ metadata domain attributes, which could encode temporal epoch, genre, or other aspects of the domain. The challenge of domain adaptation is that the meaning of features can shift across each metadata dimension: for example, the meaning of 'plant' may depend on genre (agriculture versus industry), while the meaning of 'like' may depend on epoch. To account for this, the feature embeddings should smoothly shift over domain graphs, such as the one shown in Figure 1; this would allow us to isolate the domain general aspects of each feature. Related settings have been considered only for supervised domain adaptation, where some labeled data is available in each domain (Joshi et al., 2013), but not in the unsupervised case.

More formally, we assume each instance $n$ is augmented with a vector of $M$ binary domain attributes, $\boldsymbol{z}_n \in \{0,1\}^M$. These attributes may overlap, so that we could have an attribute for the epoch 1800-1849, and another for the epoch 1800-1899. We define $z_{n,0} = 1$ as a shared attribute, which is active for all instances. We capture domain shift by estimating embeddings $\mathbf{h}_i^{(m)}$ for each feature $i$ crossed with each domain attribute $m$. We then compute the embedding for each instance by summing across the relevant domain attributes, as shown

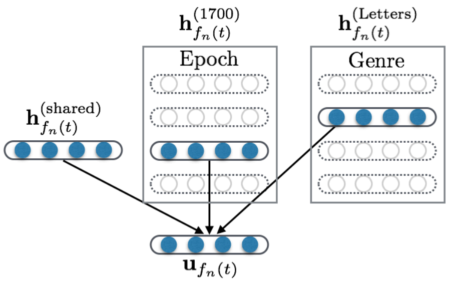

$\mathbf{h}^{(1700)}_{f_n(t)}$
Epoch
$\mathbf{h}^{(\text{Letters})}_{f_n(t)}$
Genre
$\mathbf{h}^{(\text{shared})}_{f_n(t)}$
$\mathbf{u}_{f_n(t)}$

Figure 3: Aggregating multiple embeddings.

in Figure 3. The local "input" feature embedding $\mathbf{u}_{f_n(t)}$ is then defined as the summation, $\mathbf{u}_{f_n(t)} = \sum_{m=0}^{M} z_{n,m}\mathbf{h}^{(m)}_{f_n(t)}$.

The role of the global embedding $\mathbf{h}_i^{(0)}$ is to capture domain-neutral information about the feature $i$, while the other embeddings capture attribute-specific information. The global feature embeddings should therefore be more robust to domain shift, which is "explained away" by the attribute-specific embeddings. We therefore use only these embeddings when constructing the augmented representation, $\boldsymbol{x}_n^{(\text{aug})}$. To ensure that the global embeddings capture all of the domain-general information about each feature, we place an L2 regularizer on the attribute-specific embeddings. Note that we do not learn attribute-specific "output" embeddings $\mathbf{v}$; these are shared across all instances, regardless of domain.

The attribute-based embeddings yield a new training objective for instance $n$,

$$\begin{aligned}\ell_n =& \frac{1}{T}\sum_{t=1}^{T}\sum_{t'\neq t}^{T}\Bigg[\log\sigma([\sum_{m=0}^{M} z_{n,m}\mathbf{h}^{(m)}_{f_n(t)}]^\top \mathbf{v}_{f_n(t')})\\ &+k\mathbb{E}_{i\sim P^{(n)}_{t'}}\log\sigma(-[\sum_{m=0}^{M} z_{n,m}\mathbf{h}^{(m)}_{f_n(t)}]^\top \mathbf{v}_i)\Bigg]. \quad (2)\end{aligned}$$

For brevity, we omit the regularizer from Equation 2. For feature $f_n(t)$, the (unregularized) gradients of $\mathbf{h}^{(m)}_{f_n(t)}$ and $\mathbf{v}_{f_n(t')}$ *w.r.t* $\ell_{n,t}$ are

$$\begin{aligned}\frac{\partial \ell_{n,t}}{\mathbf{h}^{(m)}_{f_n(t)}} =& \frac{1}{T}\sum_{t'\neq t}^{T} z_{n,m}\Big[(1-\sigma(\mathbf{u}^\top_{f_n(t)}\mathbf{v}_{f_n(t')}))\mathbf{v}_{f_n(t')}\\ &-k\mathbb{E}_{i\sim P^{(n)}_{t'}}\sigma(\mathbf{u}^\top_{f_n(t)}\mathbf{v}_i)\mathbf{v}_i\Big] \quad (3)\end{aligned}$$

$$\frac{\partial \ell_{n,t}}{\mathbf{v}_{f_n(t')}} = \frac{1}{T}\sum_{t'\neq t}^{T}(1-\sigma(\mathbf{u}^\top_{f_n(t)}\mathbf{v}_{f_n(t')}))\mathbf{u}_{f_n(t)}. \quad (4)$$

For each feature $i$ drawn from the noise distribution $P^{(n)}_{t'}$, the gradient of $\mathbf{v}_i$ *w.r.t* $\ell_{n,t}$ is

$$\frac{\partial \ell_{n,t}}{\mathbf{v}_i} = -\frac{1}{T}\sigma(\mathbf{u}^\top_{f_n(t)}\mathbf{v}_i)\mathbf{u}_{f_n(t)}. \quad (5)$$

## 4 Experiments

We evaluate FEMA on part-of-speech (POS) tagging, in two settings: (1) adaptation of English POS tagging from news text to web text, as in the SANCL shared task (Petrov and McDonald, 2012); (2) adaptation of Portuguese POS tagging across a graph of related domains over several centuries and genres, from the Tycho Brahe corpus (Galves and Faria, 2010). These evaluations are complementary: English POS tagging gives us the opportunity to evaluate feature embeddings in a well-studied and high-impact application; Portuguese POS tagging enables evaluation of multi-attribute domain adaptation, and demonstrates the capability of our approach in a morphologically-rich language, with a correspondingly large number of part-of-speech tags (383). As more historical labeled data becomes available for English and other languages, we will be able to evaluate feature embeddings and related techniques there.

### 4.1 Implementation details

While POS tagging is classically treated as a structured prediction problem, we follow Schnabel and Schütze (2014) by taking a classification-based approach. Feature embeddings can easily be used in feature-rich sequence labeling algorithms such as conditional random fields or structured perceptron, but our pilot experiments suggest that with sufficiently rich features, classification-based methods can be extremely competitive on these datasets, at a fraction of the computational cost. Specifically, we apply a support vector machine (SVM) classifier,

| Component | Feature template |
|---|---|
| Lexical (5) | $w_{i-2} = X, w_{i-1} = Y, \ldots$ |
| Affixes (8) | $X$ is prefix of $w_i$, $\|X\| \leq 4$<br>$X$ is suffix of $w_i$, $\|X\| \leq 4$ |
| Orthography (3) | $w_i$ contains number, uppercase character, or hyphen |

Table 1: Basic feature templates for token $w_i$.

adding dense features from FEMA (and the alternative representation learning techniques) to a set of basic features.

### 4.1.1 Basic features

We apply sixteen feature templates, motivated by by Ratnaparkhi (1996). Table 1 provides a summary of the templates; there are four templates each for the prefix and suffix features. Feature embeddings are learned for all lexical and affix features, yielding a total of thirteen embeddings per instance. We do not learn embeddings for the binary orthographic features. Santos and Zadrozny (2014) demonstrate the utility of embeddings for affix features.

### 4.1.2 Competitive systems

We consider three competitive unsupervised domain adaptation methods. Structural Correspondence Learning (Blitzer et al., 2006, SCL) creates a binary classification problem for each pivot feature, and uses the weights of the resulting classifiers to project the instances into a dense representation. Marginalized Denoising Autoencoders (Chen et al., 2012, mDA) learn robust representation across domains by reconstructing pivot features from artificially corrupted input instances. We use structured dropout noise, which has achieved state-of-art results on domain adaptation for part-of-speech tagging (Yang and Eisenstein, 2014). We also directly compare with WORD2VEC[3] word embeddings, and with a "no-adaptation" baseline in which only surface features are used.

### 4.1.3 Parameter tuning

All the hyperparameters are tuned on development data. Following Blitzer et al. (2006), we consider pivot features that appear more than 50 times in all the domains for SCL and mDA. In SCL, the parameter $K$ selects the number of singular vectors of the projection matrix to consider; we try values between 10 and 100, and also employ feature normalization and rescaling. For embedding-based methods, we choose embedding sizes and numbers of negative samples from $\{25, 50, 100, 150, 200\}$ and $\{5, 10, 15, 20\}$ respectively. The noise distribution $P_t^{(n)}$ is simply the unigram probability of each feature in the template $t$. Mikolov et al. (2013b) argue for exponentiating the unigram distribution, but we find it makes little difference here. The window size of word embeddings is set as $5$. As noted above, the attribute-specific embeddings are regularized, to encourage use of the shared embedding $\mathbf{h}^{(0)}$. The regularization penalty is selected by grid search over $\{0.001, 0.01, 0.1, 1.0, 10.0\}$. In general, we find that the hyperparameters that yield good word embeddings tend to yield good feature embeddings too.

## 4.2 Evaluation 1: Web text

Recent work in domain adaptation for natural language processing has focused on the data from the shared task on Syntactic Analysis of Non-Canonical Language (SANCL; Petrov and McDonald, 2012), which contains several web-related corpora (newsgroups, reviews, weblogs, answers, emails) as well as the WSJ portion of OntoNotes corpus (Hovy et al., 2006). Following Schnabel and Schütze (2014), we use sections 02-21 of WSJ for training and section 22 for development, and use 100,000 unlabeled WSJ sentences from 1988 for learning representations. On the web text side, each of the five target domains has an unlabeled training set of 100,000 sentences (except the ANSWERS domain, which has 27,274 unlabeled sentences), along with development and test sets of about 1000 labeled sentences each. In the spirit of truly unsupervised domain adaptation, we do not use any target domain data for parameter tuning.

**Settings** For FEMA, we consider only the single-embedding setting, learning a single feature embedding jointly across all domains. We select 6918 pivot features for SCL, according to the method described above; the final dense representation is produced by performing a truncated singular value decomposition on the projection matrix that arises from the

[3] https://code.google.com/p/word2vec/

| Target | baseline | MEMM | SCL | mDA | word2vec | FLORS | FEMA |
|---|---|---|---|---|---|---|---|
| NEWSGROUPS | 88.56 | 89.11 | 89.33 | 89.87 | 89.70 | 90.86 | **91.26** |
| REVIEWS | 91.02 | 91.43 | 91.53 | 91.96 | 91.70 | **92.95** | 92.82 |
| WEBLOGS | 93.67 | 94.15 | 94.28 | 94.18 | 94.17 | 94.71 | **94.95** |
| ANSWERS | 89.05 | 88.92 | 89.56 | 90.06 | 89.83 | 90.30 | **90.69** |
| EMAILS | 88.12 | 88.68 | 88.42 | 88.71 | 88.51 | 89.44 | **89.72** |
| AVERAGE | 90.08 | 90.46 | 90.63 | 90.95 | 90.78 | 91.65 | **91.89** |

Table 2: Accuracy results for adaptation from WSJ to Web Text on SANCL dev set.

| Target | baseline | MEMM | SCL | mDA | word2vec | FLORS | FEMA |
|---|---|---|---|---|---|---|---|
| NEWSGROUPS | 91.02 | 91.25 | 91.51 | 91.83 | 91.35 | 92.41 | **92.60** |
| REVIEWS | 89.79 | 90.30 | 90.29 | 90.95 | 90.87 | **92.25** | 92.15 |
| WEBLOGS | 91.85 | 92.32 | 92.32 | 92.39 | 92.42 | 93.14 | **93.43** |
| ANSWERS | 89.52 | 89.74 | 90.04 | 90.61 | 90.48 | 91.17 | **91.35** |
| EMAILS | 87.45 | 87.77 | 88.04 | 88.11 | 88.28 | 88.67 | **89.02** |
| AVERAGE | 89.93 | 90.28 | 90.44 | 90.78 | 90.68 | 91.53 | **91.71** |

Table 3: Accuracy results for adaptation from WSJ to Web Text on SANCL test set.

weights of the pivot feature predictors. The mDA method does not include any such matrix factorization step, and therefore generates a number of dense features equal to the number of pivot features. Memory constraints force us to choose fewer pivots, which we achieve by raising the threshold to 200, yielding 2754 pivot features.

**Additional systems** Aside from SCL and mDA, we compare against published results of FLORS (Schnabel and Schütze, 2014), which uses distributional features for domain adaptation. We also republish the baseline results of Schnabel and Schütze (2014) using the Stanford POS Tagger, a maximum entropy Markov model (MEMM) tagger.

**Results** As shown in Table 2 and 3, FEMA outperforms competitive systems on all target domains except REVIEW, where FLORS performs slightly better. FLORS uses more basic features than FEMA; these features could in principle be combined with feature embeddings for better performance. Compared with the other representation learning approaches, FEMA is roughly 1% better on average, corresponding to an error reduction of 10%. Its training time is approximately 70 minutes on a 24-core machine, using an implementation based on gensim.[4] This is slightly faster than SCL, although slower than mDA with structured dropout noise.

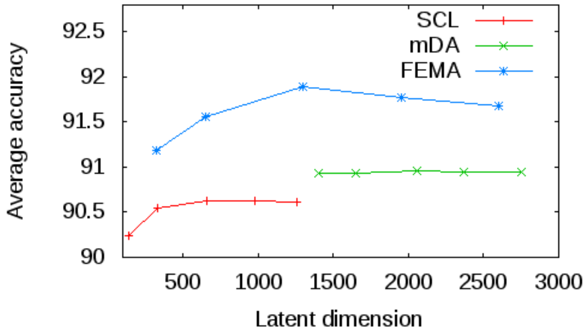


Figure 4: Accuracy results with different latent dimensions on SANCL dev sets.

Figure 4 shows the average accuracy on the SANCL development set, versus the latent dimensions of different methods. The latent dimension of SCL is modulated by the number of singular vectors; we consider sizes 10, 25, 50, 75, and 100. In mDA, we consider pivot feature frequency thresholds 500, 400, 300, 250, and 200. For FEMA, we consider embedding sizes 25, 50, 100, 150, and 200. The resulting latent dimensionality multiplies these sizes by the number of non-binary templates

[4] http://radimrehurek.com/gensim/

| Task | baseline | SCL | mDA | word2vec | FEMA | |
|---|---|---|---|---|---|---|
| | | | | | single embedding | attribute embeddings |
| from 1800-1849 | | | | | | |
| → 1750 | 88.74 | 89.31 | 90.11 | 89.24 | 90.25 | **90.59** |
| → 1700 | 89.97 | 90.41 | 91.39 | 90.51 | 91.61 | **92.03** |
| → 1650 | 85.94 | 86.76 | 87.69 | 86.22 | 87.64 | **88.12** |
| → 1600 | 86.21 | 87.65 | 88.63 | 87.41 | 89.39 | **89.77** |
| → 1550 | 88.92 | 89.92 | 90.79 | 89.85 | 91.47 | **91.78** |
| → 1500 | 85.32 | 86.82 | 87.64 | 86.60 | 89.29 | **89.89** |
| AVERAGE | 87.52 | 88.48 | 89.37 | 88.30 | 89.94 | **90.36** |
| from 1750-1849 | | | | | | |
| → 1700 | 94.37 | 94.60 | 94.86 | 94.60 | 95.14 | **95.22** |
| → 1650 | 91.49 | 91.78 | 92.52 | 91.85 | 92.56 | **93.26** |
| → 1600 | 91.92 | 92.51 | 93.14 | 92.83 | 93.80 | **93.89** |
| → 1550 | 92.75 | 93.21 | 93.53 | 93.21 | **94.23** | 94.20 |
| → 1500 | 89.87 | 90.53 | 91.31 | 91.48 | 92.05 | **92.95** |
| AVERAGE | 92.08 | 92.53 | 93.07 | 92.80 | 93.56 | **93.90** |

Table 4: Accuracy results for adaptation in the Tycho Brahe corpus of historical Portuguese.

13. FEMA dominates the other approaches across the complete range of latent dimensionalities. The best parameters for SCL are dimensionality $K = 50$ and rescale factor $\alpha = 5$. For both FEMA and WORD2VEC, the best embedding size is 100 and the best number of negative samples is 5.

### 4.3 Evaluation 2: Historical Portuguese

Next, we consider the problem of *multi-attribute domain adaptation*, using the Tycho Brahe corpus of historical Portuguese text (Galves and Faria, 2010), which contains syntactic annotations of Portuguese texts in four genres over several centuries (Figure 1). We focus on temporal adaptation: training on the most modern data in the corpus, and testing on increasingly distant historical text.

**Settings** For FEMA, we consider domain attributes for 50-year temporal epochs and genres; we also create an additional attribute merging all instances that are in neither the source nor target domain. In SCL and mDA, 1823 pivot features pass the threshold. Optimizing on a source-domain development set, we find that the best parameters for SCL are dimensionality $K = 25$ and rescale factor $\alpha = 5$. The best embedding size and negative sample number are 50 and 15 for both FEMA and WORD2VEC.

**Results** As shown in Table 4, FEMA outperforms competitive systems on all tasks. The column "single embedding" reports results with a single feature embedding per feature, ignoring domain attributes; the column "attribute embeddings" shows that learning feature embeddings for domain attributes further improves performance, by 0.3-0.4% on average.

## 5 Similarity in the embedding space

The utility of word and feature embeddings for POS tagging task can be evaluated through word similarity in the embedding space, and its relationship to type-level part-of-speech labels. To measure the label consistency between each word and its top $Q$ closest words in the vocabulary we compute,

$$\text{Consistency} = \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{Q} \beta(w_i, c_{ij})}{|V| \times Q} \tag{6}$$

where $|V|$ is the number of words in the vocabulary, $w_i$ is the $i$-th word in the vocabulary, $c_{ij}$ is the $j$-th closest word to $w_i$ in the embedding space (using cosine similarity), $\beta(w_i, c_{ij})$ is an indicator function that is equal to 1 if $w_i$ and $c_{ij}$ have the same most common part-of-speech in labeled data.

We compare feature embeddings of different templates against WORD2VEC embeddings. All embeddings are trained on the SANCL data, which is

| Embedding | $Q = 5$ | $Q = 10$ | $Q = 50$ | $Q = 100$ |
|---|---|---|---|---|
| WORD2VEC | 47.64 | 46.17 | 41.96 | 40.09 |
| FEMA-current | 68.54 | 66.93 | 62.36 | 59.94 |
| FEMA-prev | 55.34 | 54.18 | 50.41 | 48.39 |
| FEMA-next | 57.13 | 55.78 | 52.04 | 49.97 |
| FEMA-all | 70.63 | 69.60 | 65.95 | 63.91 |

Table 5: Label consistency of the $Q$-most similar words in each embedding. FEMA-all is the concatenation of the current, previous, and next-word FEMA embeddings.

also used to obtain the most common tag for each word. Table 5 shows that the FEMA embeddings are more consistent with the type-level POS tags than WORD2VEC embeddings. This is not surprising, since they are based on feature templates that are specifically designed for capturing syntactic regularities. In simultaneously published work, Ling et al. (2015) present "position-specific" word embeddings, which are an alternative method to induce more syntactically-oriented word embeddings.

Table 6 shows the most similar words for three query keywords, in each of four different embeddings. The next-word and previous-word embeddings are most related to syntax, because they help to predict each other and the current-word feature; the current-word embedding brings in aspects of orthography, because it must help to predict the affix features. In morphologically rich languages such as Portuguese, this can help to compute good embeddings for rare inflected words. This advantage holds even in English: the word 'toughness' appears only once in the SANCL data, but the FEMA-current embedding is able to capture its morphological similarity to words such as 'tightness' and 'thickness'. In WORD2VEC, the lists of most similar words tend to combine syntax and topic information, and fail to capture syntactic regularities such as the relationship between 'and' and 'or'.

## 6 Related Work

**Representation learning** Representational differences between source and target domains can be a major source of errors in the target domain (Ben-David et al., 2010). To solve this problem, cross-domain representations were first induced via auxiliary prediction problems (Ando and Zhang, 2005), such as the prediction of *pivot features* (Blitzer et al., 2006). In these approaches, as well as in later work on denoising autoencoders (Chen et al., 2012), the key mechanism is to learn a function to predict a subset of features for each instance, based on other features of the instance. Since no labeled data is required to learn the representation, target-domain instances can be incorporated, revealing connections between features that appear only in the target domain and features that appear in the source domain training data. The design of auxiliary prediction problems and the selection of pivot features both involve heuristic decisions, which may vary depending on the task. FEMA avoids the selection of pivot features by directly learning a low-dimensional representation, through which features in each template predict the other templates.

| | |
|---|---|
| **'new'** | |
| FEMA-current | nephew, news, newlywed, newer, newspaper |
| FEMA-prev | current, local, existing, international, entire |
| FEMA-next | real, big, basic, local, personal |
| WORD2VEC | current, special, existing, newly, own |
| **'toughness'** | |
| FEMA-current | tightness, trespass, topless, thickness, tenderness |
| FEMA-prev | underside, firepower, buzzwords, confiscation, explorers |
| FEMA-next | aspirations, anguish, pointers, organisation, responsibilities |
| WORD2VEC | parenting, empathy, ailment, rote, nerves |
| **'and'** | |
| FEMA-current | amd, announced, afnd, anesthetized, anguished |
| FEMA-prev | or, but, as, when, although |
| FEMA-next | or, but, without, since, when |
| WORD2VEC | but, while, which, because, practically |

Table 6: Most similar words for three queries, in each embedding space.

An alternative is to link unsupervised learning in the source and target domains with the label distribution in the source domain, through the framework of posterior regularization (Ganchev et al., 2010). This idea is applied to domain adaptation by Huang and Yates (2012), and to cross-lingual learning by Ganchev and Das (2013). This approach requires a forward-backward computation for representation learning, while FEMA representations can be learned without dynamic programming, through negative sampling.

**Word embeddings** Word embeddings can be viewed as special case of representation learning, where the goal is to learn representations for each word, and then to supply these representations in place of lexical features. Early work focused on discrete clusters (Brown et al., 1990), while more recent approaches induce dense vector representations; Turian et al. (2010) compare Brown clusters with neural word embeddings from Collobert and Weston (2008) and Mnih and Hinton (2009). Word embeddings can also be computed via neural language models (Mikolov et al., 2013b), or from canonical correlation analysis (Dhillon et al., 2011). Xiao and Guo (2013) induce word embeddings across multiple domains, and concatenate these representations into a single feature vector for labeled instances in each domain, following EasyAdapt (Daumé III, 2007). However, they do not apply this idea to unsupervised domain adaptation, and do not work in the structured feature setting that we consider here. Bamman et al. (2014) learn geographically-specific word embeddings, in an approach that is similar to our multi-domain feature embeddings, but they do not consider the application to domain adaptation. We can also view the distributed representations in FLORS as a sort of word embedding, computed directly from rescaled bigram counts (Schnabel and Schütze, 2014).

Feature embeddings are based on a different philosophy than word embeddings. While many NLP features are lexical in nature, the role of a word towards linguistic structure prediction may differ across feature templates. Applying a single word representation across all templates is therefore suboptimal. Another difference is that feature embeddings can apply to units other than words, such as character strings and shape features. The tradeoff is that feature embeddings must be recomputed for each set of feature templates, unlike word embeddings, which can simply be downloaded and plugged into any NLP problem. However, computing feature embeddings is easy in practice, since it requires only a light modification to existing well-optimized implementations for computing word embeddings.

**Multi-domain adaptation** The question of adaptation across multiple domains has mainly been addressed in the context of supervised multi-domain learning, with labeled data available in all domains (Daumé III, 2007). Finkel and Manning (2009) propagate classification parameters across a tree of domains, so that classifiers for sibling domains are more similar; Daumé III (2009) shows how to induce such trees using a nonparametric Bayesian model. Dredze et al. (2010) combine classifier weights using confidence-weighted learning, which represents the covariance of the weight vectors. Joshi et al. (2013) formulate the problem of multi-attribute multi-domain learning, where all attributes are potential distinctions between domains; Wang et al. (2013) present an approach for automatically partitioning instances into domains according to such metadata features. Our formulation is related to multi-domain learning, particularly in the multi-attribute setting. However, rather than partitioning all instances into domains, the domain attribute formulation allows information to be shared across instances which share metadata attributes. We are unaware of prior research on unsupervised multi-domain adaptation.

## 7 Conclusion

Feature embeddings can be used for domain adaptation in any problem involving feature templates. They offer strong performance, avoid practical drawbacks of alternative representation learning approaches, and are easy to learn using existing word embedding methods. By combining feature embeddings with metadata domain attributes, we can perform domain adaptation across a network of interrelated domains, distilling the domain-invariant essence of each feature to obtain more robust representations.

**Acknowledgments** This research was supported by National Science Foundation award 1349837. We thank the reviewers for their feedback. Thanks also to Hal Daumé III, Chris Dyer, Slav Petrov, and Djamé Seddah.

## References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853.

David Bamman, Chris Dyer, and Noah A. Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 828–834, Baltimore, MD.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 120–128.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 440–447, Prague.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1990. Class-Based N-Gram models of natural language. *Computational Linguistics*, 18:18–4.

Minmin Chen, Z. Xu, Killian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 160–167.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the Association for Computational Linguistics (ACL)*, Prague.

Hal Daumé III. 2009. Bayesian multitask learning with latent hierarchies. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 135–142. AUAI Press.

Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. 2011. Multi-view learning of word embeddings via cca. In *Advances in Neural Information Processing Systems*, pages 199–207.

Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1-2):123–149.

E. Eaton, M. Desjardins, and T. Lane. 2008. Modeling transfer relationships between learning tasks for improved inductive transfer. *Machine Learning and Knowledge Discovery in Databases*, pages 317–332.

Jenny R. Finkel and Christopher Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 602–610, Boulder, CO.

Charlotte Galves and Pablo Faria. 2010. Tycho Brahe Parsed Corpus of Historical Portuguese. `http://www.tycho.iel.unicamp.br/~tycho/corpus/en/index.html`.

Kuzman Ganchev and Dipanjan Das. 2013. Cross-lingual discriminative learning of sequence models with posterior regularization. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 1996–2006.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, Seattle, WA.

Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *The Journal of Machine Learning Research*, 13(1):307–361.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 57–60, New York, NY.

Fei Huang and Alexander Yates. 2012. Biased representation learning for domain adaptation. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 1313–1323.

Mahesh Joshi, Mark Dredze, William W. Cohen, and Carolyn P. Rosé. 2013. What's in a domain? multi-domain learning for multi-attribute data. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 685–690, Atlanta, GA.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, CO.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Neural Information Processing Systems (NIPS)*, pages 1041–1048.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Neural Information Processing Systems (NIPS)*, pages 1081–1088.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, volume 59.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 133–142.

Cicero D. Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1818–1826.

Tobias Schnabel and Hinrich Schütze. 2014. Flors: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association of Computational Linguistics*, 2:51–62.

Noah A Smith. 2011. Linguistic structure prediction. *Synthesis Lectures on Human Language Technologies*, 4(2):1–274.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representation: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 384–394, Uppsala, Sweden.

Di Wang, Chenyan Xiong, and William Yang Wang. 2013. Automatic domain partitioning for multi-domain learning. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 869–873.

Min Xiao and Yuhong Guo. 2013. Domain adaptation for sequence labeling tasks with a probabilistic language adaptation model. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 293–301.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.