The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Few Shot Transfer Learning Between Word Relatedness and Similarity Tasks Using a Gated Recurrent Siamese Network

James O' Neill, Paul Buitelaar
Insight Centre for Data Analytics
National University of Ireland, Galway
{james.oneill, paul.buitelaar}@insight-centre.org

Abstract

Word similarity and word relatedness are fundamental to natural language processing and more generally, understanding how humans relate concepts in semantic memory. A growing number of datasets are being proposed as evaluation benchmarks, however, the heterogeneity and focus of each respective dataset makes it difficult to draw plausible conclusions as to how a unified semantic model would perform. Additionally, we want to identify the transferability of knowledge obtained from one task to another, within the same domain and across domains. Hence, this paper first presents an evaluation and comparison of eight chosen datasets tested using the best performing regression models. As a baseline, we present regression models that incorporate both lexical features and word embeddings to produce consistent and competitive results compared to the state of the art. We present our main contribution, the best performing model across seven of the eight datasets - a *Gated Recurrent Siamese Network* that learns relationships between lexical word definitions. A parameter transfer learning strategy is employed for the Siamese Network. Subsequently, we present a secondary contribution which is the best performing non-sequential model: an Inductive and Transductive Transfer Learning strategy for transferring decision trees within a Random Forest to a target task that is learned from only few instances. The method involves measuring semantic distance between hidden factored matrix representations of decision tree traversal matrices.

Introduction

The task of measuring word similarity and word association/relatedness is a fundamental natural language task that has direct impact on upstream challenges such as information extraction, taxonomy generation, natural language understanding and generation etc. The distinction between association and similarity has been highlighted and well studied in previous work (Hill, Reichart, and Korhonen 2016). It is also an active area of research in neuroscience and cognitive science (Patterson, Nestor, and Rogers 2007), studying the complex relationships between concept representations in semantic distributed neural networks and how action selectivity in neuron activity occurs for different symbolic relationships in the frontal cortex. This phenomena has been modeled from text by utilizing lexical resources (Alvarez and Lim 2007; Hughes and Ramage 2007) and distributional semantic models (DSMs) for distance and similarity estimation (Bollegala, Matsuo, and Ishizuka 2007) by manually constructing a semantic network (i.e humanly annotated resources) for how humans might perceive concepts to be related or similar, akin to selectivity of neurons in the same regions of the *Visual Word Form Area* (VWFA) given a particular task (e.g association or similarity). Likewise, this paper begins with an evaluation of both lexical and embedding similarity features across eight (one similarity-based and seven relatedness-based) datasets on ten best performing regressors as a baseline. We then select the top performing models and propose a method for transferring instances, parameters and features from one task to another, within the same domain and across domains only using few instances from the target task. Secondly, a *Gated Recurrent Siamese Network* (*GRU-SN*) for pairwise learning of word definitions is presented. In addition, the more generalizable 1^{st} hidden layer is transferred across the *GRU-SN's* between the related domains, while the second layer is tuned but constrained to the weight distribution of the source task, $\mathcal{T}_s$. All results are compared against the SoTA for each dataset, while also proposing a new baseline for transfer learning (TL) on these benchmark datasets.

Dataset Descriptions

Recently, a number of datasets have been established as gold standard evaluations for association/relatedness and similarity. However, many of these datasets focus on varying aspects of similarity and relatedness e.g different distributions of part of speech (PoS), concept concreteness, and task focus (i.e relatedness, similarity or both). We start with a brief overview of these datasets, splitting both the descriptions into subsections based on relatedness/association and true similarity.

Relatedness/Association Based Datasets

SimVerb is a verb similarity dataset that was created by Gerz et al. (Kipper et al. 2008) by leveraging VerbNet and its verb class extensions, consisting of a test set of 3000 pairs and a development set of 500 verb pairs. There are 827 unique verbs (by lemma) from 29 VerbNet Levin classes. Verb lemmas are split into 4 categories based on frequency provided by the British National Corpus[1] (BNC). The annotator agreement is $\bar{\rho}_p = 0.84$ for 843 native speakers (65,000 ratings in total) and the correlation between the average score of a rater and the remaining raters is $\rho_p = 0.86$. The best performing distributional model uses a dependency based skip-gram model with negative sampling with a projection layer dimension $h_{proj} = 500$, achieving a $\rho_s = 0.389$ (ρ_s denotes Spearman Correlation) on the development set and $\rho_s = 0.351$ on the test set. Bruni et al. (Bruni, Tran, and Baroni 2011) introduced **BLESS**, a dataset for comparing distributional models for semantic relatedness and compositionality. *BLESS* consists of 200 concrete nouns of various classes, paired with related nouns, verbs or adjectives, with different relation types (e.g hypernymy, meronymy) along with random relations. This makes up 26,553 pairs in total, the largest dataset analysed in this paper. The **Semantic Neighbors** (SN) dataset (Panchenko 2013) includes 14,682 pairs of 462 nouns, half are randomly chosen and others are synonym pairs retrieved from *WordNet* and a synonym database, complementing *BLESS* since it does not contain synonym pairs. Finkelstein et al's. (Finkelstein et al. 2001) **WordSim353** semantic similarity dataset has been considered as a benchmark for semantic similarity, consisting of 353 word pairs in English with 13 human annotated similarity judgments. The original work focused on vector representations where the dimensions of a vector represent the frequency of a word across 27 different domains, with the intuition that words with similar frequencies across domains are related. Similarly to our work, they also use *WordNet* measures to account for word relations and use a combination of this with the vector of domain word frequencies. This was the basis for which WordSim353 was created. The **MEN** dataset is one of the largest datasets in this section, consisting of 3000 word pairs with relatedness scores (0-1] created by online raters. MEN was created (Bruni, Tran, and Baroni 2014) to benchmark the evaluation of multi-modal models. Word pairs were sampled from a combined corpora *ukWaC* and *Wackypedia* if the frequency $f \geqslant 700$, providing good coverage over the degree of relatedness. Two of the authors annotated all 3000 pairs in random order which produced $p = 0.68$ and also had an average correlation with the annotators of $\bar{\rho}_s = 0.84$. Similarity is tested on a development set (33% of dataset) to test the generalization, in our evaluation we report 10-fold cross validation results to ensure full coverage of all pairs. The **Rareword** dataset (Luong, Socher, and Manning 2013) includes 2034 morphologically complex and subsequently rare word pairs with the average similarity rating from 10 raters, originally used to build DSMs by splitting words into morpheme units and passing these units separately as input vectors to an RNN encoder achieving $\rho_s = 0.436$, which we consider as a baseline for subsequent evaluation of single task and few shot transfer results.

Similarity Dataset

RG65 The Rubenstein and Goodenough (RG) (Rubenstein and Goodenough 1965) dataset was first introduced in 1965 with the purpose of answering the question, "are words that are synonymous more likely to be contextually similar ?". They address this question and also conjecture that words are synonymous if the contexts overlap, "however, is apparently uncertain since words of low or medium synonymy differ relatively little in overlap.". The annotator agreement for 15 people was $\rho_p = 0.82$ (ρ_p denotes Pearson Correlation). To date, the current SoTA for this similarity dataset between these 65 word pairs is by Camacho et al. (Camacho-Collados, Pilehvar, and Navigli 2015) who more recently used a knowledge-based *Wiktionary* approach using a method called *Align, Disambiguate and Walk*, achieving $\rho_s = 0.92$. **SimLex-999** is a gold standard resource introduced by Hill et al. (Hill, Reichart, and Korhonen 2016) for the evaluation of models for conceptual word meaning, instead of relatedness or association. They make a clear and important distinction between association/relatedness and true similarity with varying concrete concepts and abstract concepts on verb, noun and adjective pairs, in comparison to WordSim353 that does not account for this in the similarity scoring guidelines, motivating the need for models of real similarity. Authors point out that semantic models performance is significantly lower in comparison to other gold standard evaluation datasets due to its diversity and instances where similarity and association are distinctly different across many PoS, reasoning that *Simlex* provides a good platform for testing and evaluating different semantic models. Additionally, since *Simlex* is the only dataset that we deem to truly account for similarity, we focus our TL efforts on only transferring from relatedness/association domains to *Simlex*. Hill et al. (Hill, Reichart, and Korhonen 2016) have tested the `skipgram` model (trained on Wikipedia) which achieves $\rho_s = 0.37$, (Mikolov et al. 2013) only trained on running monolingual text $\rho_s = 0.56$ (Schwartz, Reichart, and Rappoport 2015), knowledge-based approach (*WordNet*, *Framenet* etc.), a model that uses rich paraphrase data (Wieting et al. 2015) achieving $\rho_s = 0.68$, features from various word embeddings and two lexical databases, $\rho_s = 0.76$ (Recski, Iklódi, and Pajkossy 2016).

Siamese Networks

Siamese networks (SN's) are neural networks that learn relationships by encoding pairs of objects to separately measure the distance/similarity between encoded representations of both networks. SN's have been applied to learning sentence relatedness scores and textual entailment (Mueller and Thyagarajan 2016) achieving SoTA results on the *SICK* semantic textual similarity task. In the computer vision community, Hoffer et. al (Hoffer and Ailon 2015) have also trained a triplet network that takes three input images, provides l_2 distance between the output of three separate but identical CNNs ($f(x_+, x), (x, x_-)$ where x and x_+ are the same class and x_- are not) for digit recognition and animal/vehicle image classification. Chopra et al. (Chopra, Hadsell, and LeCun 2005) have used a contrastive loss function to learn parameters for a function so that similar points are drawn together and dissimilar points repulse. This motivates the idea of using contrastive loss for learning similarity and relatedness in the classification setting, where the

[1] http://www.natcorp.ox.ac.uk/

emphasis is put on similar or related word pairs to be drawn closer. Intuitively, encoded representations of paired word definitions might only have few dimensions of their encoded sentence representations that are close, but many potentially distant given the context in which a definition describes a word. In this sense, we consider the definitions of words to be more similar if only a few dimensions are very similar even if the remainder are distant, to account for more subtle relations and similarities. This requires that the range of annotation scores be converted to classes, in our approach binary classes. In order to choose negative and positive classes for $\mathcal{T}_{rel} \in \mathcal{D}_{rel}$ and $\mathcal{T}_{sim} \in \mathcal{D}_{sim}$, we analyze the distribution of normalized annotated scores ([0-1]) and choose the threshold for classes to be $\bar{y}$ for each respective dataset. This hypothesis is compared to MSE for the original regression task and Negative Log-Likelihood (NLL) in classification.

Transfer Learning

In a large number of problems in machine learning (ML) we do not have enough labeled data (or poorly labeled) for a particular task but we have labeled data for a related task $\mathcal{T}_r$ in the same domain $\mathcal{D}_s$ in space $\mathcal{X} \in \mathbb{R}^d$, or a related domain $\mathcal{D}_t$. Concretely, when $P(Y|X)$ changes between training and testing, also referred to as *model adaptation*. Humans often learn with little or no knowledge of unfamiliar observations by applying knowledge of related and/or similar observations. This kind of learning has alluded many modern ML models as a meta learning procedure (when suitable) whilst only relying on large amounts of data as a cumbersome replacement. Although, in the past decade there has been increasing interest for learning (Bengio 2012) that makes use of available knowledge that is transferable. Supervised TL can be broadly split into two types: *Transductive Transfer Learning* (TTL) and *Inductive Transfer Learning* (ITL). In TTL, we transfer knowledge across different domains, $\mathcal{D}_s \rightarrow \mathcal{D}_t$ where feature spaces $\mathcal{X}_s \nsim \mathcal{X}_t$, proving to be a more difficult transfer learning scenario given the change in domain. In the ITL setting, the target task $\mathcal{T}_t$ is different from the source task $\mathcal{T}_s$, however both $[\mathcal{T}_s, \mathcal{T}_t] \in \mathcal{D}$, requiring a mapping function $f(\theta)$. This can be achieved using four approaches: instance transfer where we learn to transfer whole instances X_i that are related across tasks, transferring features across tasks, transferring parameters across tasks or knowledge based transfer (e.g using lexical resources like WordNets semantic network to transfer knowledge). We primarily focus on both parameter transfer and relational-knowledge transfer for our *Siamese Network*. However, we also present an alternative ensemble model that incorporates instance, feature and parameter transfer in a unified manner. -

Methodology

All eight datasets used in the experiments are those described in Section *Dataset Description*, making up 49, 045 word pairs (predominantly from BLESS and SN). This also includes a large number of definition sentence pairs retrieved from *Wiktionary* which we evaluate the *GRU-SN* model on.

Non-sequential Regression Models

Lexical graph measures from *WordNet* such as *Leacock & Chodorow* (LCH) similarity , *Wu & Palmer* (WUP) similarity and path similarity are used. LCH similarity computes the shortest path length (SPL) for a word pair and normalizes the value by the *WordNet* hierarchy maximum path length (SPL) (computed using the WUP similarity). LCH is then the inverse of the SPL between the words. We also use pretrained GoogleNews *Word2Vec* embeddings by comparing the cosine similarity between word pair embeddings, using this value as a feature for predicting scores. The combination of both semantic knowledge based measures and co-occurrence based predict vectors (i.e `skipgram` vectors) is motivated by previous literature in cognitive science. Representing words and their context words without any external influence or knowledge can be problematic, also known as the *symbol grounding problem*. Hence, we attempt to mitigate the lack of external knowledge when only using co-occurrence based measures, particularly for similarity tasks like *Simlex*.

Baseline Model Configurations A number of well founded models are considered for regression as an initial baseline, the first of which is the *Least Absolute Shrinkage and Selection Operator* (LASSO) regression model. The LASSO model uses an MSE loss with a regularization term $\alpha|\theta|l_1$ to penalize large coefficients. ElasticNet is also considered, by optimizing both $\mathcal{L}_1$ and $\mathcal{L}_2$ norms. The Huber regression model optimizes MSE for samples where $|(y - X^T\theta)/\sigma| \leqslant \epsilon \geqslant |(y - X^T\theta)/\sigma|$ where σ ensures that the loss is scale invariant to y. All linear regression models are transferred using a weighted average θ_s $\forall$ $\mathcal{T}_s \in \mathcal{D}_s \rightarrow \mathcal{T}_t \in \mathcal{D}_t$ where $\mathcal{T}_{1:s} \nsim \mathcal{T}_t$ are assigned based on the euclidean distance to W_t for $\mathcal{T}_t$ by first computing the Pointwise Mutual Information (PMI) between coefficients θ_s of source models M_s and coefficients θ_t of target model, M_t. The Support Vector Machine (SVM) uses a 2^{nd} degree RBF with $\gamma = 0.001$, $\epsilon = 0.1$ and $C = 1.0$ and shrinkage. The Gradient Boosted Tree Regression (GBTR) model uses a learning rate $\alpha = 0.001$, 10 decision trees, a tuned depth of $max(d_G) = 5$ and uses a least squares (LS) loss function. For both Bagging and Random Forest (RF) regressors, 10 *C4.5* trees are used with MSE to measure the split quality with a maximum depth $\max(d_G) = 5$. For the k-nearest neighbor regressor we find best results are obtained when $k = 5$ with standard euclidean distance. These models are considered as a baseline along with the aforementioned SoTA in the previous section for both single task learning and few shot transfer learning.

Random Forest Few Shot Transfer Learning To answer the questions of what, where and how to transfer in both the ITL (transfer in domains) and TTL (transfer across domains) setting, a strategy for decision tree transfer from RF's is described. We look to transfer individual trees $F \in \mathcal{F}$, for two situations: $\mathcal{T}_s \in \mathcal{D}_{rel} \rightarrow \mathcal{T}_t \in \mathcal{D}_{sim}$ and $\mathcal{T}_s \in \mathcal{D}_{rel} \rightarrow \mathcal{T}_t \in \mathcal{D}_{rel}$. This is a form of feature-based transfer, since RF's bootstrap sample for $x \in X$ and features $f \in \mathcal{F} \in \mathbb{R}^N$

and model transfer since we transfer weighted *C4.5* trees. We exploit the decision paths taken for a set of sampled instances $X_s \in X \forall \mathcal{F}$ and then compare lower dimensional factored representations f_h from $\mathcal{T}_s \rightarrow \mathcal{T}_t$. Concretely, the attributes are indexed (w2v=1,path=2,..) and appear as decisions statements, we represent each instance X_i as a vector of decisions it is passed through in the decision tree (DT) where elements are ordered decision values. In this sense, we are comparing *C4.5* trees with labeled leaves with the same features $\forall X_s \in \mathcal{X}_s$ in an attempt to identify transferable *C4.5* trees. Frameworks for semantic similarity between DT's have been explored before (Ntoutsi, Kalousis, and Theodoridis 2008), although in our case we are also comparing DT's from different distributions from different domains $X_s \in \mathcal{D}_s \rightarrow X_t \in \mathcal{D}_t$. Segev et al. (Segev et al. 2017) approached this problem by adjusting the target DT using a combination of expanding/reducing the source tree structure and replacing the template decision values by inserting the target DT values. In our approach we too seek to encode and find similarity in structure between DT's from X_s and X_t, although we want to do this for the similarity between latent representations of trees and discard any trees in an RF that are too distant from the DT trained on a subsample for $\mathcal{T}_t$. Furthermore, latent representations that account for weak correlations (e.g subtle correlations between underlying relatedness and similarity tasks) are also a desirable characteristic when comparing decision trees. Therefore, we propose the use of *Correlation Explanation* (CorEx) (Ver Steeg and Galstyan 2014) - an unsupervised method used to find latent factors h_f that can discover hierarchical representations of X, and scales well for high dimensional data, such as a feature vector of decision values. We choose latent factors $|h_f| = 20$ and then compute the euclidean norm between two latent factor vectors that represent trees, one constructed from subsamples in an RF for $\mathcal{T}_s$ and a single decision tree learned from a 10 % subsample of *Simlex*, $\mathcal{T}_t$ (also the case for TTL with *GRU-SN*). RF is then transferred from $\mathcal{T}_s \in \mathcal{D}_s \rightarrow \mathcal{T}_t \in \mathcal{D}_t$. An element $\mathcal{M}_{ij}$ indicates a path j along the tree that has been visited by instance X_i for all trees in RF, returning a list of vectors of latent factors for each decision tree which can be represented in a matrix M_{γ_s} where γ are the latent factors for all trees. Equation 1 shows the multivariate mutual information (MMI) between a set of decision paths $M_{dp}^{(i)} \in M^{(i)}$ and latent factors h_f. Here we aim to maximize both for each decision path that is represented as matrix $\mathcal{M} \in \mathbb{R}^N$, so to find factors h_f that best explain the correlations in $M^{(i)}$. In Equation 2, α is randomly initialized.

$$MMI(M_{dp}; h_f) = \textstyle\sum_{i \in \mathbb{R}_n} I(M_{dp}^{(i)} : h_f) - I(M_p^{(i)} : h_f) \quad (1)$$

$$\max_{\alpha, h_f | M_{dp}} \sum_{j=1}^{m} \sum_{i=1}^{n} \alpha_{i,j} I(h_{f_j} : M_{dp}^{(i)}) - \sum_{j=1}^{m} I(h_{f_j} : M_{dp}^{(i)}) \quad (2)$$

The euclidean norm $\forall C4.5 \in RF$ to a target *C4.5* tree trained on few examples on a $\mathcal{T}_t$ where the source trees are weighted by their latent factor euclidean distance to h_f representation of $\mathcal{T}_t$, for *ITL* where we identify loosely related the tasks for transfer, and *TTL*.

Gated Recurrent Siamese Network

An alternative way to combine word vector and semantic network based similarities is to extract informative definitions for each word pair to use as input pairs $\langle s_1^{(i)}, s_2^{(i)} \rangle \in [\mathbb{N} \rightarrow \mathbb{R}]$ to the *GRU-SN*. The definitions are retrieved *Wiktionary* which also provides a dictionary containing a range of other properties such as PoS tags, etymologies, antonyms, synonyms etc. An example of word pairs (impatient, anxious) for *Simlex* is the following - ("Restless and intolerant of delays" - "Full of anxiety or disquietude greatly concerned or solicitous especially respecting something future or unknown") where in classification $\phi = y \geqslant \bar{y}$, $(\phi \rightarrow 1) \wedge (\neg\phi \rightarrow 0)$. These definitions are then in the form of pretrained GoogleNews word vectors to pass as input to the *GRU-SN* with labels y as scores provided by each dataset We consider at most 4 definitions per word in *Wiktionary*. Furthermore, we count the number of definitions for all word pairs so to average across the results when computing the ρ_s and ρ_p, this avoids biased results toward pairs which have more definition pairings and also makes for a fair comparison against the non-sequential models. Equations 3, 4, 5 and 6 present the nonlinear functions of the Gated Recurrent Unit (GRU) (Chung et al. 2014), where z_t is the update gate, r_t denotes the reset gate, candidate hidden layer $\tilde{h}_t$ and hidden layer h_t, which is an interpolation between h_{t-1} and $\tilde{h}_{t-1}$. The GRU treats both input and forget gates as a single function z_t and does not contain a separate memory cell like Long Short-Term Memory (LSTM). It also merges the cell state and hidden state. Since the GRU do not use memory cells, hence the content of a cell is fully exposed to other units, instead, the GRU controls for content coming from h_{t-1}, whereas the LSTM instead controls the flow of new information from the current input x_t.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (3)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (4)$$

$$\tilde{h}_t = \tanh(W_r r_t + r_t \odot (U h_{t-1})) \quad (5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (6)$$

In preliminary experiments we found the *GRU* to outperform the *LSTM*, herein we focus on *GRU-SN*. In our case, both networks in the *GRU-SN* have hidden layers $l = 2$.

Parameter Settings The *GRU-SN* consists of a standard 2-hidden layer GRU network for each input sentence $\langle s_1^{(i)}, s_2^{(i)} \rangle \; \forall i \in N$ where N is the number of instances in a dataset X. Orthogonalization weight initialization is used to produce uncorrelated weights and gradients are clipped for $10^{-5} \leq W \geq 1 - 10^{-5}$, although a *GRU* should circumvent exploding or vanishing gradients. These weights are also tied across both of the 2-hidden layer networks where both use a dropout rate $p = 0.2$. The batch size for each X is chosen in proportion to number of N instances, a minibath between 5% - 10% (e.g SN and BLESS use 5% while smaller datasets such as *Simlex* use 10%).

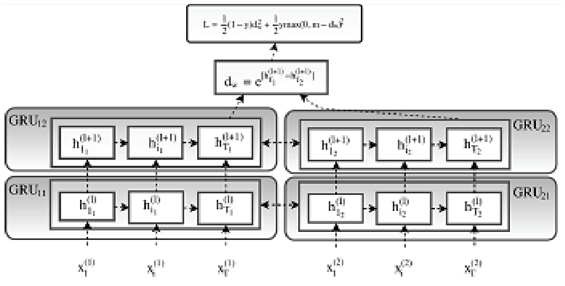

Figure 1: 2-hidden layer GRU Siamese Network

Loss Functions Apart from using MSE loss for regression and NLL in classification, we also consider *Contrastive Loss*. Contrastive loss attempts to pull weights of neighboring samples together, and push non-neighbors elements apart. Likewise, close elements of encoded representations $\langle h_T^{(l_1)}, h_T^{(l_2)} \rangle$, for word definitions input pairs $\langle s_1^{(i)}, s_2^{(i)} \rangle$ are pulled together, where T denotes the max length of a sentence set for both definition pairs (in our experiments $T = 30$). Equation (7) shows the contrastive loss function proposed by (Hadsell, Chopra, and LeCun 2006), where D_w is the parameterized distance function of $|h_T^{(l_1)} - h_T^{(l_2)}|$ (i.e definition encoding pairs) and θ is all shared network weights. This requires that we treat the normalized target label $\tilde{y}$ as $p(y|\theta, x)$ instead of a continuous rating. We assume this conversion is acceptable by making the assumption that human ratings for $\mathcal{D}_{sim}$ and $\mathcal{D}_{rel}$ are approximately close to the rater correlation scores $p_p \forall \mathcal{T} \in \mathcal{D}$, thereby interpreting $\bar{y}$ as estimating uncertainty that a given word pair are classified as related/similar or not.

$$\mathcal{L}(\theta) = \frac{1}{2}(1 - Y)D_w^2 + \frac{1}{2} Y max(0, margin - D_w)^2 \quad (7)$$

Transfer Learning for Siamese Networks

Previous work (Yosinski et al. 2014) empirically show earlier hidden layer feature representations are more generalizable to loosely related tasks for images in the computer vision domain. Likewise, we hypothesize that this is similar for embedded word representations in the lower level features from initial hidden layers, hence we look to transfer weights of the first hidden layer from a $\mathcal{T}_s \forall \mathcal{D}_{rel}$ to *Simlex*. Additionally, we enforce a prior density on target weights $[W_s^{(1)}, U_s^{(1)}, b_s^{(1)}, W_s^{(2)}, U_s^{(2)}, b_s^{(2)}]$ (denoted as θ_s) obtained from trained networks in $\mathcal{T}_s$. We take a simple approach to TTL by initializing the weights $\mathcal{T}_s$ with a probability density estimate $p(\theta_s|X_s)$ from $\mathcal{T}_t$, carried out for each respective layer, particularly in the few shot learning setting.

The methodology can be summarized as the four following steps: (1) evaluate single task regressors on all word similarity and relatedness datasets and compare to existing SoTA, (2) apply a TL approach for the best performing learner across related tasks and domains, (3) evaluate proposed *GRU-SN* trained on *Wiktionary* lexical definitions of word pairs and (4) transfer weights across tasks using source weight initialization and source weight distribution.

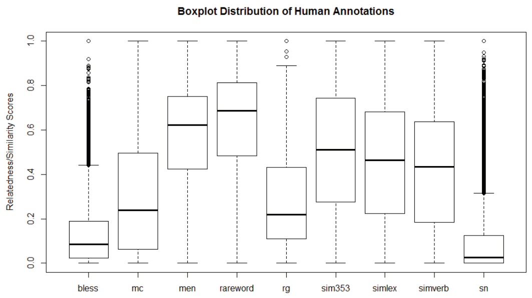

Figure 2: Boxplot of Annotation Scores

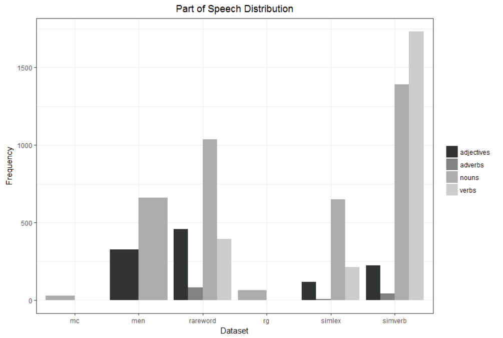

Figure 3: Part of Speech Distribution

Experimental Results

Figure 2 shows the distribution of the output space $\mathcal{Y}$ for all datasets. Evidently, the output distribution Y is similar between *Simlex*, *Simverb* and *Sim353* which gives an inclination for the subsequent transfer learning. The larger *SN* and *BLESS* datasets show low scores for similarity with a similar trend of outlying high scores for both datasets. Also, *Rareword* and *MEN* both have similar quartile ranges. These findings are also reflected in 3. For example, the output space for *MEN* and *Rareword* is somewhat reflected by the PoS frequency distrbution, as both have similar proportions of nouns and adjectives. Similarly, *Simlex* and *SimVerb* also have close annotation score quartile ranges, that is also reflected by PoS, apart from the strong focus of verbs in *SimVerb* which is not included in *Simlex*. These observations are used to help motivate our focus for a particular subset of transfer learning between datasets.

Regression Results With the exception of *MEN*, ensemble meta learning models such as Random Forests (RFs) have achieved the best 10-fold cross-validation (CV) correlations measures, as shown in Table 1 and 2 (bolded font represents best performing models), with Gradient Boosting showing close performance. The learning curve on a 10-fold CV is shown for each RF classifier in figure 4. *BLESS* shows no consistent improvement in accuracy after trained on 50 % onwards. *RG* shows high variance in results between training 75 % - 87.50 % which can be explained by the small sample

		Lasso	Elastic	Bayes	Huber
RG65	PC	0.517	0.458	0.477	**0.494**
	SC	0.494	0.441	0.452	0.486
SimVerb	PC	0.379	0.396	0.377	0.367
	SC	0.358	**0.378**	0.353	0.349
WordSim353	PC	0.688	0.598	0.611	0.633
	SC	**0.735**	0.629	0.662	0.703
Simlex	PC	0.521	0.490	0.539	0.571
	SC	0.500	0.511	0.511	**0.559**
MEN	PC	0.761	0.750	0.742	0.772
	SC	0.773	0.762	0.759	**0.786**
Rareword	PC	0.477	0.485	0.429	0.467
	SC	0.491	**0.505**	0.440	0.490
BLESS	PC	0.612	0.632	0.628	0.630
	SC	0.514	0.522	0.527	**0.545**
SN	PC	0.521	0.556	0.553	0.542
	SC	0.359	0.375	**0.384**	0.365

Table 1: Linear Regression 10-fold CV Results
(Spearman = SC, Pearson= PC)

		k-NN	ANN	GBoost	RF	Bagging	SVR
RG65	PC	0.508	0.576	0.659	0.629	0.526	0.583
	SC	0.393	0.445	**0.526**	0.602	0.496	0.557
SimVerb	PC	0.272	0.309	0.415	0.418	0.337	0.400
	SC	0.251	0.303	0.390	**0.395**	0.299	0.369
WordSim353	PC	0.568	0.361	0.649	0.661	0.480	0.571
	SC	0.567	0.341	0.667	**0.672**	0.463	0.634
Simlex	PC	0.479	0.496	0.514	0.601	0.527	**0.479**
	SC	0.471	0.461	0.499	**0.576**	0.501	0.464
MEN	PC	0.663	0.482	0.787	0.768	0.672	0.548
	SC	0.667	0.482	**0.772**	0.763	0.679	0.605
Rareword	PC	0.395	0.396	0.501	0.549	0.285	0.486
	SC	0.388	0.388	0.483	**0.531**	0.277	0.489
BLESS	PC	0.600	0.624	0.660	0.647	0.557	0.639
	SC	0.506	0.532	0.564	**0.567**	0.466	0.547
SN	PC	0.495	0.554	0.592	0.563	0.516	0.549
	SC	0.367	0.360	0.467	**0.469**	0.383	0.346

Table 2: Nonlinear Regression 10-fold CV Results

of 65 word pairs. *Simlex* shows a steady incline in performance, at 85 % of the training dataset we find a plateau in 10-CV performance, a similar trend also exhibited for *MEN*.

Table 3 shows models trained in $\mathcal{T}_{rel} \in \mathcal{D}_s$ and tested on $\mathcal{T}_{sim} \in \mathcal{D}_t$, and also $\mathcal{D}_s^{(1)} \to \mathcal{D}_s^{(2)}$. Transferring weights from *WordSim353* → *Simlex* for the *Huber* regression model has led to $\rho_s = 0.490$ which was originally $\rho_s = 0.574$ for *Simlex*. The performance for zero-shot transfer for the SVR model is also notably similar. The ensemble methods that performed the best on single tasks have not transfered as well from $\mathcal{D}_s \to \mathcal{D}_t$ compared to the *Huber* and *SVR* models. In fact, *SVR* shows almost identical results to that of the source task performance. This pattern re-occurs for the other datasets that *Transfer Across Different Domains* (*TTL*). In addition, testing *Rareword* on *WordSim353* produces the best performance for *WordSim353* thus far, this can be explained by the larger dataset but also $\mathcal{Y}$ is similar, as demonstrated in the 2 boxplot.

Table 4 shows that transferring weighted *C4.5* trees from *SimVerb, WordSim353, Rareword & MEN* (in bolded font) shows performance improvements for ρ_s when ensembled with a single *C4.5* tree trained on 90 *Simlex* instances. Notably, *SimVerb* has carried over the most knowledge to *Simlex*, also slightly improving over the zero-shot learning setting. Similarly, when *Rareword C4.5* trees are ensembled with a single *Simlex* tree there is a 0.051 point increase in p_s.

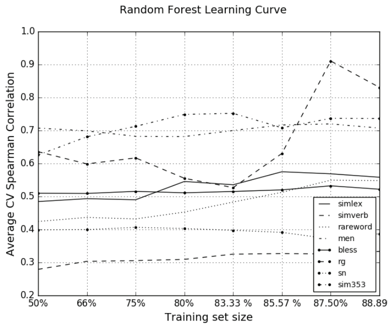

Figure 4: Random Forest Cross-Validation Learning Curve

In-Domain Transfer		Huber	GB	RF	SVR
Rareword → WordSim353	PC	0.690	0.709	0.688	0.728
	SC	**0.723**	0.727	0.718	0.730
MEN → Rareword	PC	0.542	0.542	0.548	0.549
	SC	0.551	**0.557**	0.556	0.556
Rareword → MEN	PC	0.699	0.702	0.683	0.687
	SC	0.700	0.701	0.687	**0.737**
WordSim353 → RareWord	PC	0.604	0.575	0.559	0.609
	SC	**0.622**	0.570	0.565	0.614
SN → BLESS	PC	0.633	0.655	0.665	0.608
	SC	0.543	**0.565**	0.525	0.547
Out-Of-Domain Transfer					
WordSim353 → Simlex	PC	0.490	0.413	0.447	0.487
	SC	0.471	0.422	0.412	**0.474**
SimVerb → Simlex	PC	0.476	0.333	0.445	0.544
	SC	0.472	0.261	0.392	**0.512**
SN → Simlex	PC	0.491	0.382	0.421	0.474
	SC	**0.479**	0.405	0.406	0.474
MEN → Simlex	PC	0.443	0.431	0.430	0.495
	SC	0.422	0.416	0.389	**0.479**
Rareword → Simlex	PC	0.432	0.306	0.409	0.479
	SC	0.410	0.307	0.364	**0.473**
BLESS → Simlex	PC	0.487	0.395	0.319	0.470
	SC	**0.470**	0.381	0.344	0.447
Simlex→ WordSim353	PC	0.735	0.724	0.733	0.743
	SC	0.696	0.699	0.702	**0.725**
Simlex→SimVerb	PC	0.340	0.353	0.347	0.354
	SC	0.296	0.306	0.302	**0.308**
Simlex → SN	PC	0.473	0.469	0.477	0.474
	SC	**0.378**	0.372	0.376	0.375
Simlex→MEN	PC	0.696	0.700	0.700	0.707
	SC	0.695	0.699	0.697	**0.705**
Simlex→Rareword	PC	0.566	0.566	0.557	0.561
	SC	0.554	**0.563**	0.547	0.544
Simlex→BLESS	PC	0.665	0.670	0.668	0.666
	SC	0.525	**0.526**	0.525	0.523

Table 3: Zero Shot Parameter Transfer Learning

Siamese Network Results The results from learning pairwise relationships between *Wiktionary* definitions has shown improvements. Table 5 show the number of definitions pairs for each dataset and Table 6 shows the results of the *GRU-SN* for the average 10-fold CV correlation measures, in both classification and regression. We find that there are significantly less definition pairs for *Rareword* due to the complex morphology associated with rare, infrequent words. Treating the problem as one of binary classification where we interpret scores of relatedness or similarity as probability estimates $P(\hat{y}|x_1, x_2)$ has shown an improve-

Target	*Source*			
Simlex ↤	Simlex	WordSim353	RG	SN
PC	0.567	0.551	0.421	0.581
SC	0.580	**0.565**	0.349	0.536
Simlex ↤	BLESS	SimVerb	RareWord	MEN
PC	0.319	0.581	0.612	0.579
SC	0.344	**0.594**	**0.563**	**0.560**

Table 4: Few Shot Parameter and Instance Decision Tree Transfer with *CE* Latent Factor Similarity

Frequency	Simlex	WordSim353	RG65	SN
Vocabulary	1028	437	48	5901
Definitions	145,992	40,158	3,679	648,845
	BLESS	SimVerb	RareWord	MEN
Vocabulary	8,020	826	2,946	639
Definitions	2,076,130	649,898	30,131	125,368

Table 5: Wiktionary Definition Statistics

Wiktionary Definitions		Simlex	WordSim353	RG65	SN
GRU-SNN w/ Contrastive	PC	0.691	0.784	0.555	0.632
	SC	0.657	**0.788**	0.503	0.516
GRU-SNN w/ NLL	PC	0.734	0.809	0.573	0.617
	SC	**0.671**	0.759	**0.557**	**0.594**
GRU-SNN w/ MSE	PC	0.683	0.743	0.583	0.590
	SC	0.638	0.701	0.549	0.579
		BLESS	SimVerb	Rareword	MEN
GRU-SNN w/ Contrastive	PC	0.638	0.596	0.672	0.601
	SC	**0.541**	**0.572**	0.619	**0.722**
GRU-SNN w/ NLL	PC	0.594	0.654	0.708	0.591
	SC	0.517	0.612	**0.647**	0.719
GRU-SNN w/ MSE	PC	0.603	0.524	0.369	0.592
	SC	0.528	0.510	0.479	0.645

Table 6: *GRU-SN* With Contrastive, NLL & MSE Loss

Target	*Source*			
Simlex ↤	Simlex	WordSim353	RG	SN
PC	0.691	0.633	0.382	0.596
SC	0.657	0.605	0.357	0.520
Simlex ↤	BLESS	SimVerb	RareWord	MEN
PC	0.522	0.651	0.624	0.658
SC	0.449	**0.623**	0.629	0.614

Table 7: *GRU-SN* Few Shot Parameter Transfer using *NLL*

ment for all datasets.

From Table 6 we see a significant performance increase on 10-CV results, particularly for *Rareword, Simlex* and *WordSim353*. *Simlex* has seen a significant improvement from what was $\rho_s = 0.576$ to $\rho_s = 0.691$ by exploiting the larger and more descriptive definitions. Although *RG65* has improved performance using 3,679 definition pairs, due to its small training size it is difficult to perform parameter transfer, shown by its low correlation scores in Table 7.

Table 7 shows the results of few shot learning transfer using *NLL* loss. Hence, these models are trained on $\mathcal{T}_s$ and then tuned using *Simlex* trained on a 10% subsample. Parameter transfer is carried out by fixing the 1^{st} hidden layer from $\mathcal{T}_s$ and learning only the 2^{nd} layer features from *Simlex*, initialized from kernel density estimation of the second layer weights in $\mathcal{T}_s$. Similarly to the zero shot learning setting for the alternative models, *SimVerb* has provided the best generalization properties on *Simlex*.

Discussion and Conclusion

We observe that single task learning approaches for all datasets have performed comparably to prior SoTA as described in the dataset description. RF's with 10 *C4.5* trees have outperformed other regression models on the majority of tasks. Additionally, transferring *C4.5* trees constructed from $\mathcal{T}_{rel} \rightarrow \mathcal{T}_{sim}$ (i.e *Simlex*) has performed almost as well as models only trained on $\mathcal{D}_s$. Transferring knowledge through *C4.5* trees within RF's in the few shot learning setting has led to good performance in comparison to a fully trained RF on the target task, it is also the first of its kind. Additionally, the RF transfer learning strategy has the advantage of being applicable to pairwise tasks that are not sequential in nature.

Transferring between closely related tasks is more effective as defined by the similarity between latent representations in the *GRU-SN* model, also reinforced by analysis of the similarity between output distributions $\mathcal{Y}$ and PoS distribution. The *GRU-SN* for single task learning on *Wiktionary* definitions has produced the best overall results, particularly for classification. Through experimentation of constraining the gradients using kernel density estimates for the 2^{nd} hidden layer we suggest that the standard deviation should be contingent on the distance between tasks i.e weight constraints are relaxed when tasks/domains are far apart and vice-versa. The improved average correlation measures across datasets can be attributed to the use of *Wiktionary* definition pairs, as (1) there is a considerably larger number of instances for training, (2) a unified approach to encoding lexically driven pairwise vectors, (3) an empirical experimentation of various loss function for this specific task and (4) a *TTL* strategy for knowledge transfer over identical network architectures across different domains. This highlights the commonalities between different domains although the tasks are different.

We have focused on term similarity and term relatedness as it is fundamental to how humans perceive and understand relations between concepts in the real world. However, the *GRU-SN* can be used in other problems that require learning relations between two or more sequences. We hope these findings ultimately lead to less fragmentation when comparing results across different datasets and establish new results on the transferability between domains and tasks.

References

Alvarez, M. A., and Lim, S. 2007. A graph modeling of semantic similarity between words. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, 355–362. IEEE.

Bengio, Y. 2012. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, 17–36.

Bollegala, D.; Matsuo, Y.; and Ishizuka, M. 2007. Measuring semantic similarity between words using web search engines. *www* 7:757–766.

Bruni, E.; Tran, G. B.; and Baroni, M. 2011. Distributional semantics from text and images. In *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*, 22–32. Association for Computational Linguistics.

Bruni, E.; Tran, N.-K.; and Baroni, M. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)* 49(2014):1–47.

Camacho-Collados, J.; Pilehvar, M. T.; and Navigli, R. 2015. Nasari: a novel approach to a semantically-aware representation of items. In *HLT-NAACL*, 567–577.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 539–546. IEEE.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, 406–414. ACM.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, 1735–1742. IEEE.

Hill, F.; Reichart, R.; and Korhonen, A. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

Hoffer, E., and Ailon, N. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 84–92. Springer.

Hughes, T., and Ramage, D. 2007. Lexical semantic relatedness with random graph walks. In *EMNLP-CoNLL*, 581–589.

Kipper, K.; Korhonen, A.; Ryant, N.; and Palmer, M. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation* 42(1):21–40.

Luong, T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, 104–113.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mueller, J., and Thyagarajan, A. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, 2786–2792.

Ntoutsi, I.; Kalousis, A.; and Theodoridis, Y. 2008. A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, 810–821. SIAM.

Panchenko, A. e. a. 2013. *Similarity measures for semantic relation extraction*. Ph.D. Dissertation, PhD thesis, Université catholique de Louvain & Bauman Moscow State Technical University.

Patterson, K.; Nestor, P. J.; and Rogers, T. T. 2007. Where do you know what you know? the representation of semantic knowledge in the human brain. *Nature reviews. Neuroscience* 8(12):976.

Recski, G. A.; Iklódi, E.; and Pajkossy. 2016. Measuring semantic similarity of words using concept networks. Association for Computational Linguistics.

Rubenstein, H., and Goodenough, J. B. 1965. Contextual correlates of synonymy. *Communications of the ACM* 8(10):627–633.

Schwartz, R.; Reichart, R.; and Rappoport, A. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *CoNLL*, volume 2015, 258–267.

Segev, N.; Harel, M.; Mannor, S.; Crammer, K.; and El-Yaniv, R. 2017. Learn on source, refine on target: a model transfer learning framework with random forests. *IEEE transactions on pattern analysis and machine intelligence* 39(9):1811–1824.

Ver Steeg, G., and Galstyan, A. 2014. Discovering structure in high-dimensional data through correlation explanation. In *Advances in Neural Information Processing Systems*, 577–585.

Wieting, J.; Bansal, M.; Gimpel, K.; Livescu, K.; and Roth, D. 2015. From paraphrase database to compositional paraphrase model and back. *arXiv preprint arXiv:1506.03487*.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, 3320–3328.