Multi-View Domain Adapted Sentence Embeddings for Low-Resource Unsupervised Duplicate Question Detection

Nina Poerner[1,2] **and Hinrich Schütze**[1]
[1]Center for Information and Language Processing, LMU Munich, Germany
[2]Corporate Technology Machine Intelligence (MIC-DE), Siemens AG Munich, Germany
`poerner@cis.uni-muenchen.de | inquiries@cislmu.org`

Abstract

We address the problem of Duplicate Question Detection (DQD) in low-resource domain-specific Community Question Answering forums. Our multi-view framework MV-DASE combines an ensemble of sentence encoders via Generalized Canonical Correlation Analysis, using unlabeled data only. In our experiments, the ensemble includes generic and domain-specific averaged word embeddings, domain-finetuned BERT and the Universal Sentence Encoder. We evaluate MV-DASE on the CQADupStack corpus and on additional low-resource Stack Exchange forums. Combining the strengths of different encoders, we significantly outperform BM25, all single-view systems as well as a recent supervised domain-adversarial DQD method.

1 Introduction

Duplicate Question Detection is the task of finding questions in a database that are equivalent to an incoming query. Many Community Question Answering (CQA) forums leave this task to the collective memory of their users. This results in unnecessary manual work for community members as well as delayed access to answers (Hoogeveen et al., 2015).

Automatic DQD is often approached as a supervised problem with community-generated training labels. However, smaller CQA forums may suffer from label sparsity: On Stack Exchange, 50% of forums have fewer than 160 user-labeled duplicates, and 25% have fewer than 50 (see Figure 1).[1]

To overcome this problem, two avenues have been explored: The first is supervised domain-adversarial training on a label-rich source forum (Shah et al., 2018), which works best when source and target domains are related. The second is unsupervised DQD via representation learning (Charlet and Damnati, 2017; Lau and Baldwin, 2016), which requires only unlabeled questions. In this paper, we take the unsupervised avenue.

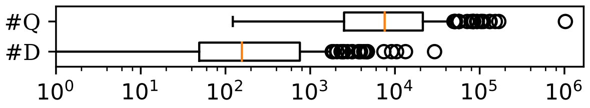

Figure 1: Distribution (log-scale box plot) of number of questions (#Q) and number of labeled duplicates (#D) on Stack Exchange. $N = 165$ forums.

A major challenge in the context of domain-specific CQA forums is that language usage may differ from the "generic" domains of existing representations. To illustrate this point, compare the following Nearest Neighbor lists of the word "tree", based either on generic GloVe embeddings (Pennington et al., 2014) or on FastText embeddings (Bojanowski et al., 2017) that were trained on specific CQA forums:

generic (GloVe): trees, branches, leaf
chess: searches, prune, modify
outdoors: trees, trunk, trunks
gis: strtree, rtree, btree
wordpress: trees, hierachy, hierarchial
gaming: trees, treehouse, skills

Charlet and Damnati (2017) and Lau and Baldwin (2016) report that representations trained on in-domain data perform better on unsupervised DQD than generic representations. But in a low-resource setting, the amount of unlabeled in-domain data is limited. This can result in low coverage or quality, as illustrated by the in-domain embedding neighbors of "tree" in the smallest forum from our dataset:

windowsphone: dreamspark, l535ds, generally

[1] `archive.org/details/stackexchange` [data dump: 2018-12-20]

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 1630–1641,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

	generic	**domain-specific**
contextualized	f_G: GloVe	f_D: FastText (in-domain)
noncontextualized	f_U: USE	f_B: BERT (domain-finetuned)

Table 1: Ensemble used in our experiments.

It is therefore desirable to combine the overall quality and coverage of generic representations with the domain-specificity of in-domain representations via multi-view learning. There is a large body of work on multi-view word embeddings (see Section 2.3), including domain adapted word embeddings (Sarma et al., 2018).

Recent representation learning techniques go beyond the word level and embed larger contexts (e.g., sentences) jointly (Peters et al., 2018; Devlin et al., 2019; Cer et al., 2018). To reflect this paradigm shift, we take multi-view representation learning from the word to the sentence level and propose MV-DASE (Multi-View Domain Adapted Sentence Embeddings), a framework that combines an ensemble of sentence encoders via Generalized Canonical Correlation Analysis (see Section 3.1).

MV-DASE uses **unlabeled in-domain data only**, making it applicable to the problem of unsupervised DQD. As a framework, it is **agnostic** to the internal specifics of its ensemble. In Section 3.2, we describe an ensemble of different sentence encoders: domain-specific and generic, contextualized and noncontextualized (see Table 1). In Sections 4 and 5, we demonstrate that MV-DASE is effective at **duplicate retrieval** on the CQADupStack corpus (Hoogeveen et al., 2015) and on additional low-resource Stack Exchange forums. **Significance tests** show significant gains over BM25, all single-view systems and domain-adversarial supervised training as proposed by Shah et al. (2018). In Sections 6 and 7, we successfully evaluate MV-DASE on **two additional benchmarks**: the SemEval-2017 DQD shared task (Nakov et al., 2017) as well as the unsupervised STS Benchmark (Cer et al., 2017).

2 Related Work

2.1 Duplicate Question Detection

Most prior work on DQD (e.g., Bogdanova et al. (2015); Dos Santos et al. (2015); Baldwin et al. (2016); Zhang et al. (2017); Rodrigues et al. (2017); Hoogeveen et al. (2018)) focuses on supervised architectures. As discussed, these approaches are not applicable to forums with few or no labeled duplicates.

Shah et al. (2018) tackle label sparsity by domain-adversarial training (ADA). More specifically, they train a bidirectional Long-Short Term Memory Network (LSTM) (Hochreiter and Schmidhuber, 1997) on a label-rich source forum, while minimizing the distance between source and target domain representations. Their approach beats BM25 and a simple transfer baseline in cases where source and target domain are closely related (e.g., *AskUbuntu→SuperUser*), but not on more distant pairings. This is not ideal, as the existence of a big related source forum is not guaranteed.

An alternative is unsupervised DQD via representation learning, which does not require any labels. Charlet and Damnati (2017) use a word embedding-based soft cosine distance for duplicate ranking. In a recent DQD shared task (SemEval-2017 task 3B, Nakov et al. (2017)), their best unsupervised system trails the best supervised system by only 2% Mean Average Precision (MAP). This seems reasonable, given that the implicit objective of many representation learning methods (similar representations for similar objects) is closely related to the notion of a duplicate.

Charlet and Damnati (2017) report overall better results when embeddings are trained on domain-specific data rather than Wikipedia. However, they make no attempts to combine the two domains. Lau and Baldwin (2016) evaluate two representation learning techniques (doc2vec (Le and Mikolov, 2014) and word2vec (Mikolov et al., 2013a)) on CQADupStack. They also report better results when representations are learned on domain-specific rather than generic data.

2.2 Sentence embeddings and STS

Unsupervised DQD is related to the task of unsupervised Semantic Textual Similarity (STS), i.e., sentence similarity scoring (Cer et al., 2017). Arora et al. (2017) show that a weighted average over pre-trained word embeddings, followed by principal component removal, is a strong baseline for STS. We use their weighting scheme, Smooth Inverse Frequency (SIF), in Section 3.2.

Averaged word embeddings are insensitive to word order. This stands in contrast to contextualized encoders, such as LSTMs or Transformers (Vaswani et al., 2017). Contextualized encoders are typically trained as unsupervised language models (Peters et al., 2018; Devlin et al., 2019) or on supervised transfer tasks (Conneau et al., 2017; Cer et al., 2018). At the time of writing, weighted averaged word embeddings achieve better results than contextualized encoders on unsupervised STS.[2]

2.3 Multi-view word embeddings

Multi-view representation learning is an umbrella term for methods that transform different representations of the same entities into a common space. In NLP, it has typically been applied to word embeddings. A famous example is the cross-lingual projection of word embeddings (Mikolov et al., 2013b; Faruqui and Dyer, 2014). Monolingually, Rastogi et al. (2015) use Generalized Canonical Correlation Analysis (GCCA) to project different word representations into a common space. Yin and Schütze (2016) combine word embeddings by concatenation, truncated Singular Value Decomposition and linear projections; Bollegala and Bao (2018) use autoencoders. Sarma et al. (2018) correlate generic and domain-specific word embeddings by Canonical Correlation Analysis (CCA).

All of these methods are post-training, i.e., they are applied to fully trained word embeddings. MV-DASE falls into the same category, albeit at the sentence level (see Section 3.1). Other methods, which we will call in-training, encourage the alignment of embeddings during training (e.g., Bollegala et al. (2015); Yang et al. (2017)).

2.4 Multi-view sentence embeddings

Multi-view sentence embeddings are less frequently explored than multi-view word embeddings. One exception is Tang and de Sa (2019), who train a recurrent neural network and an average word embedding encoder jointly on an unlabeled corpus. This method is in-training, i.e., it cannot be used to combine pre-existing encoders.

Kiela et al. (2018) dynamically integrate an ensemble of word embeddings into a task-specific LSTM. They require labeled data and the resulting embeddings are task-specific.

Sarma et al. (2018) marry domain-adapted word embeddings (see Section 2.3) with InferSent (Conneau et al., 2017), a bidirectional LSTM sentence encoder trained on Stanford Natural Language Inference (SNLI) (Bowman et al., 2015). They initialize InferSent with the adapted embeddings and then retrain it on SNLI. Note that this approach is not feasible when the training regime of an encoder cannot be reproduced, e.g., when the original training data is not publicly available.

3 Method

We now describe MV-DASE as a general framework. For details on the ensemble used in this paper, see Section 3.2.

3.1 Framework

GCCA basics. Given zero-mean random vectors $\mathbf{x}_1 \in \mathbb{R}^{d_1}, \mathbf{x}_2 \in \mathbb{R}^{d_2}$, Canonical Correlation Analysis (CCA) finds linear transformations $\boldsymbol{\theta}_1 \in \mathbb{R}^{d_1}, \boldsymbol{\theta}_2 \in \mathbb{R}^{d_2}$ such that $\boldsymbol{\theta}_1^T \mathbf{x}_1$ and $\boldsymbol{\theta}_2^T \mathbf{x}_2$ are maximally correlated. Bach and Jordan (2002) show that CCA reduces to a generalized eigenvalue problem. A generalized eigenvalue problem finds scalar-vector pairs $(\rho, \boldsymbol{\theta})$ that satisfy $\mathbf{A}\boldsymbol{\theta} = \rho\mathbf{B}\boldsymbol{\theta}$ for matrices $\mathbf{A}, \mathbf{B}$. Here, $\mathbf{A}, \mathbf{B}$ are the following block matrices:

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{2,1} & \mathbf{0} \end{bmatrix} \boldsymbol{\theta} = \rho \begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{2,2} \end{bmatrix} \boldsymbol{\theta} \tag{1}$$

where $\boldsymbol{\Sigma}_{1,1}, \boldsymbol{\Sigma}_{2,2}$ are the covariance matrices of $\mathbf{x}_1, \mathbf{x}_2$ and $\boldsymbol{\Sigma}_{1,2}, \boldsymbol{\Sigma}_{2,1}$ are their cross-covariance matrices. We stack all d eigenvectors into an operator $\boldsymbol{\Theta} \in \mathbb{R}^{d \times d_1 + d_2}$. Using this operator, multi-view representations are projected as:

$$\mathbf{x}_{\text{mv}} = \boldsymbol{\Theta} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{2}$$

Generalized CCA (GCCA) generalizes CCA to three or more random vectors $\mathbf{x}_1 \dots \mathbf{x}_J$. There are several variants of GCCA (Kettenring, 1971); we follow Bach and Jordan (2002) and solve a multi-view version of Equation 1:

$$\begin{aligned} &\begin{bmatrix} \mathbf{0} & \boldsymbol{\Sigma}_{\dots} & \boldsymbol{\Sigma}_{1,J} \\ \boldsymbol{\Sigma}_{\dots} & \mathbf{0} & \boldsymbol{\Sigma}_{\dots} \\ \boldsymbol{\Sigma}_{J,1} & \boldsymbol{\Sigma}_{\dots} & \mathbf{0} \end{bmatrix} \boldsymbol{\theta} \\ = \rho &\begin{bmatrix} \boldsymbol{\Sigma}_{1,1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\dots} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{J,J} \end{bmatrix} \boldsymbol{\theta} \end{aligned} \tag{3}$$

For stability, we add $\tau\sigma_j\mathbf{I}_j$ to every covariance matrix $\boldsymbol{\Sigma}_{j,j}$, where τ is a hyperparameter (here: $\tau = 0.1$), $\mathbf{I}_j$ is the identity matrix and σ_j is the

[2] http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark

average variance of $\mathbf{x}_j$. Like in the two-view case, we stack all d eigenvectors into an operator: $\mathbf{\Theta} \in \mathbb{R}^{d \times \sum_j d_j}$.

GCCA application. Assume that we have an ensemble of J sentence encoders. The j'th encoder is denoted $f_j : \mathbb{S} \rightarrow \mathbb{R}^{d_j}$, where $\mathbb{S}$ is the set of all possible in-domain strings (here: in-domain questions) and d_j is determined by f_j. Assume also that we have a sample from $\mathbb{S}$, i.e., a corpus of unlabeled in-domain strings, denoted $S = \{s_1, \ldots, s_N\}$. From this corpus, we create one training matrix $\mathbf{X}_j$ per encoder:

$$\mathbf{X}_j \in \mathbb{R}^{N \times d_j} = \begin{bmatrix} — & f_j(s_1) & — \\ & \vdots & \\ — & f_j(s_N) & — \end{bmatrix} \tag{4}$$

From $\mathbf{X}_j$ we estimate mean vector $\bar{\mathbf{x}}_j \in \mathbb{R}^{d_j}$, covariance matrix $\mathbf{\Sigma}_{j,j} \in \mathbb{R}^{d_j \times d_j}$ and cross-covariance matrices $\mathbf{\Sigma}_{j,j'} \in \mathbb{R}^{d_j \times d_{j'}}$. We then apply GCCA as described before, yielding $\mathbf{\Theta} \in \mathbb{R}^{d \times \sum_j d_j}$. The multi-view embedding of a new input q (e.g., a test query) is:

$$f_{\text{mv}}(q) = \mathbf{\Theta} \begin{bmatrix} f_1(q) - \bar{\mathbf{x}}_1 \\ \vdots \\ f_J(q) - \bar{\mathbf{x}}_J \end{bmatrix} \tag{5}$$

3.2 Ensemble

We use MV-DASE on the following ensemble:

- weighted averaged generic GloVe vectors (Pennington et al., 2014)
- weighted averaged domain-specific FastText vectors (Bojanowski et al., 2017)
- Universal Sentence Encoder (USE) (Cer et al., 2018)
- domain-finetuned BERT (Devlin et al., 2019)

In this section, we describe the encoders in detail. Note that the choice of encoders is orthogonal to the framework and other resources could be used. Where possible, we base our selection on the literature: We choose USE over InferSent due to better performance on STS (Perone et al., 2018), and BERT over ELMo (Peters et al., 2018) due to better performance on linguistic probing tasks (Liu et al., 2019a). The choice of GloVe for generic word embeddings is based on Sarma et al. (2018).

	f_G (GloVe)	f_D (FastText)	f_B (BERT)
no SIF	.089	.083	.134
wiki SIF	.128	.100	.159
in-domain SIF	.147	.104	.176

	f_B (BERT)	ELMo
generic	.138	.103
domain-finetuned	.176	.155

Table 2: Mean Average Precision (MAP) averaged over heldout forums. Top: MAP as a function of whether and where SIF weights are estimated. Bottom: MAP of generic vs. domain-finetuned BERT and ELMo. Evaluation setup is as described in Section 4, using four heldout forums. Gray: best in column.

Weighted averaged word embeddings. We denote generic and domain-specific word embeddings of some word type i as $\mathbf{w}_{G,i} \in \mathbb{R}^{d_G}$ and $\mathbf{w}_{D,i} \in \mathbb{R}^{d_D}$. For $\mathbf{w}_{G,i}$, we use pre-trained 300-d GloVe vectors.[3] $\mathbf{w}_{D,i}$ are trained using skipgram FastText[4] (100-d, default parameters) on the in-domain corpus S. We SIF-weight all word embeddings by $a \cdot (a + p(i))^{-1}$, where $p(i)$ is the unigram probability of the word type and the smoothing factor (here: $a = 10^{-3}$) is taken from Arora et al. (2017). We find that probabilities estimated on S produce better results than the Wikipedia-based probabilities provided by Arora et al. (2017) (see Table 2, top), hence this is what we use below. After weighting, we perform top-3 principal component removal on the embedding matrices, which is beneficial for word-level similarity tasks (Mu et al., 2018). We denote the new embeddings of word type i as $\hat{\mathbf{w}}_{G,i}, \hat{\mathbf{w}}_{D,i}$. The embedding of a tokenized string $s = (s_1, \ldots, s_T)$ is computed by averaging:

$$f_G(s) = \frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{w}}_{G,s_t} \quad f_D(s) = \frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{w}}_{D,s_t}$$

Contextualized encoders. USE and BERT are downloaded as pre-trained models.[5] [6] USE is a Transformer trained on SkipThought (Kiros et al., 2015), conversation response prediction (Henderson et al., 2017) and SNLI. It outputs a single 512-d sentence embedding, which we use as-is. Below, USE is denoted f_U.

[3] nlp.stanford.edu/data/glove.42B.300d.zip

[4] github.com/facebookresearch/fastText

[5] tfhub.dev/google/universal-sentence-encoder-large/3

[6] tfhub.dev/google/bert_uncased_L-12_H-768_A-12/1

BERT is a Transformer that was pre-trained as a masked language model with next sentence prediction. We find that domain-finetuning BERT on S results in improvements over generic BERT (see Table 2, bottom). Note that domain-finetuning refers to unsupervised training as a masked language model, i.e., we only require unlabeled data (Howard and Ruder, 2018). We use default parameters[7] except for a reduced batch size of 8.

At test time, we take the following approach: BERT segments a token sequence $s = (s_1, \dots, s_T)$ into a subword sequence $s' = ([\text{CLS}], s'_1, \dots, s'_{T'}, [\text{SEP}])$, where [CLS] and [SEP] are special tokens that were used during pre-training, and $T' \geq T$. BERT produces one 768-d vector $\mathbf{v}_{l,t}$ per subword s'_t and layer $l \in [1, \dots, L]$, where L is the total number of layers (here: 12). We SIF-weight all vectors according to the probability of their subword (estimated on S) and average over layers and subwords, excluding the special tokens:

$$f_B(s) = \frac{1}{T' \cdot L} \sum_{t=1}^{T'} \sum_{l=1}^{L} \frac{a}{a + p(s'_t)} \mathbf{v}_{l,t}$$

4 Evaluation on Stack Exchange

4.1 Data

Corpora. We evaluate MV-DASE on the CQADupStack corpus (Hoogeveen et al., 2015), which is based on a 2014 Stack Exchange dump. CQADupStack contains 12 forums that have enough duplicates for supervised training; as a consequence, it may not be representative of low-resource domains. We therefore supplement it with 12 low-resource forums from the 2018-12-20 Stack Exchange dump.[8] For our purposes, low-resource means a forum with 100–200 duplicates, which we consider sufficient for evaluation but insufficient for supervised training. All duplicates in the datasets were labeled by unpaid community members. As a result, false negatives (i.e., unflagged duplicates) are common in the gold standard (Hoogeveen et al., 2016). While we do not explicitly filter for language, the vast majority of the data is in English.

		forum	#Q	#D	#T
CQADupStack forums	and	android	23697	1579	2.4M
	eng	english	41791	3506	3.4M
	gam	gaming	46896	2207	4.0M
	gis	gis	38522	1099	4.6M
	mat	mathematica	17509	1271	2.6M
	phy	physics	39355	1769	6.1M
	prg	programmers	33052	1538	5.6M
	sta	stats	42921	890	7.2M
	tex	tex	71090	4939	7.4M
	uni	unix	48454	1648	5.5M
	web	webmasters	17911	1143	2.0M
	wor	wordpress	49146	719	5.6M
low-resource forums	bud	buddhism	5350	120	670K
	che	chess	4539	154	500K
	cog	cogsci	5687	126	800K
	law	law	11059	126	1.7M
	net	networkengineering	11386	154	1.5M
	out	outdoors	4651	124	580K
	pro	productivity	2508	127	380K
	rev	reverseengineering	15619	119	790K
	sit	sitecore	5605	130	680K
	spo	sports	4531	127	430K
	sqa	sqa	8360	166	950K
	win	windowsphone	3490	192	290K

Table 3: Forum statistics. #Q: total number of questions, #D: number of labeled duplicates, #T: number of tokens in training set S. Gray: heldout forums.

Data split. We split every forum into a test and training set, such that the test set contains all duplicates and the training set contains the remaining unlabeled questions.[9] The unlabeled training set is used for FastText training, BERT domain-finetuning, SIF weight estimation and GCCA. Test queries are never seen during training, not even in an unsupervised way. For hyperparameter choices, we hold out two high-resource and two low-resource forums (highlighted in Table 3). They are not used for the final evaluation and significance tests.

Preprocessing. Every question object consists of a title (average length 9 words), a body (average length 125 words), any number of answers or comments, and metadata (e.g., upvotes, view counts). We preprocess the data with the CQADupStack package.[10] To calculate question representations, we use the concatenation of question title and body. We always ignore answers, comments and metadata, as this information is not usually available at the time a question is posted.

[7] github.com/google-research/bert/blob/master/run_pretraining.py

[8] Preprocessed low-resource data can be downloaded here: github.com/npoe/lowresourcecqa

[9] We do not use the official CQADupStack train / test split, as it is meant for supervised training and has comparatively few duplicates per test set. Since MV-DASE is unsupervised, we can afford a more robust evaluation on all labeled duplicates.

[10] github.com/D1Doris/CQADupStack

			heldout		test forums										MAP	AUC(.05)	NDCG	P@3	R@3
CQADupStack forums			and	eng	gam	gis	mat	prg	phy	sta	tex	uni	web	wor	average over test forums				
	1	BM25	.175	.162	.310	.264	.132	.119	.216	.212	.116	.171	.103	.171	.181	.821	.314	.067	.196
	2	f_G (GloVe)	.121	.093	.202	.148	.056	.084	.152	.153	.063	.120	.085	.093	.115	.755	.233	.042	.123
	3	f_D (FastText)	.123	.083	.211	.169	.079	.091	.172	.175	.085	.136	.084	.107	.131	.817	.261	.047	.138
	4	f_U (USE)	.183	.113	.347	.156	.081	.146	.195	.165	.071	.142	.110	.117	.153	.832	.285	.056	.163
	5	f_B (BERT)	.141	.129	.262	.196	.103	.099	.190	.179	.090	.135	.109	.134	.150	.805	.276	.055	.159
	6	MV-DASE	.211	.177	.371	.274	.149	.181	.259	.236	.135	.206	.145	.183	.214	.904	.362	.080	.232
	7	InferSent	.069	.047	.145	.123	.041	.041	.105	.121	.042	.078	.053	.072	.082	.667	.182	.029	.085
	8	doc2vec	.102	.057	.141	.150	.064	.069	.138	.170	.067	.125	.083	.111	.112	.799	.234	.040	.116
	9	ELMo	.141	.116	.251	.179	.081	.097	.184	.182	.087	.147	.097	.117	.142	.835	.274	.051	.149
	10	word-level CCA	.149	.109	.253	.202	.101	.111	.190	.189	.096	.156	.103	.125	.153	.851	.290	.055	.161
	11	upper bound	1.00															.351	.999
low-resource forums			bud	che	cog	law	net	out	pro	rev	sit	spo	sqa	win	average over test forums				
	12	BM25	.276	.195	.269	.345	.167	.373	.196	.186	.430	.465	.275	.349	.306	.842	.461	.116	.345
	13	f_G (GloVe)	.249	.125	.209	.312	.103	.260	.110	.134	.237	.363	.166	.239	.213	.781	.359	.079	.234
	14	f_D (FastText)	.142	.064	.132	.255	.111	.168	.067	.101	.243	.239	.173	.136	.163	.767	.314	.060	.180
	15	f_U (USE)	.332	.247	.384	.458	.152	.513	.214	.144	.282	.448	.221	.244	.306	.880	.470	.119	.352
	16	f_B (BERT)	.261	.173	.221	.335	.137	.348	.171	.143	.324	.489	.194	.257	.262	.812	.411	.099	.294
	17	MV-DASE	.378	.259	.384	.447	.184	.495	.233	.241	.427	.523	.289	.352	.358	.924	.524	.137	.407
	18	InferSent	.154	.073	.117	.236	.079	.194	.089	.078	.192	.312	.123	.161	.158	.701	.281	.054	.162
	19	doc2vec	.133	.058	.117	.239	.146	.145	.057	.080	.192	.141	.140	.090	.135	.759	.279	.048	.143
	20	ELMo	.222	.140	.228	.332	.137	.248	.136	.092	.247	.433	.171	.278	.230	.837	.387	.084	.252
	21	word-level CCA	.260	.142	.237	.325	.146	.274	.111	.186	.312	.327	.218	.194	.233	.844	.391	.086	.254
	22	ADA	.229	.164	.161	.250	.132	.207	.117	.147	.225	.299	.193	.218	.195	.823	.347	.068	.201
	23	upper bound	1.00															.341	.999

Table 4: Main results. Left: MAP on individual forums (heldout and test forums). Rightmost five columns: all metrics averaged over test forums (excluding heldout forums). Gray: best in column.

4.2 Evaluation and Metrics

Given a test query q, we rank all candidates $c \neq q$ from the same forum by $\cos(f(q), f(c))$, where f is an encoder (e.g., MV-DASE). Our metrics are MAP, AUC(.05), Normalized Discounted Cumulative Gain (NDCG), Recall@3 (R@3) and Precision@3 (P@3). AUC(.05), the area under the ROC curve up to a false positive rate of .05, is used by Shah et al. (2018). Note that upper bounds on P@3 and R@3 are not 1, since most duplicates have only one original and a few have more than three.

4.3 Baselines

Unsupervised. Our IR baseline is BM25 (Robertson et al., 1995) as implemented in Elasticsearch 6.5.4 (Gormley and Tong, 2015) with default parameters. We test against all single-view encoders from our ensemble. The remaining unsupervised baselines are:

- ELMo (Peters et al., 2018).[11] We treat ELMo like BERT (Section 3.2), i.e., we finetune[12] the language model on the in-domain corpus (3 epochs, batch size 8), SIF-weight all vectors according to in-domain word probability and then average over layers and tokens.
- Doc2vec (Le and Mikolov, 2014) trained on the in-domain corpus, using the best DQD hyperparameters reported in Lau and Baldwin (2016).
- InferSent V.1.[13] (Conneau et al., 2017)
- Our re-implementation of domain-adapted CCA word embeddings (Sarma et al. (2018), see Section 2.3). We use the same word embeddings, SIF weights and component removal described in Section 3.2. Denoted "word-level CCA" below.

[11] `tfhub.dev/google/elmo/2`

[12] `github.com/allenai/bilm-tf/blob/master/bin/restart.py`

[13] `github.com/facebookresearch/InferSent`

	MAP	AUC(.05)
1	MV-DASE, $\neg f_G$, $\neg f_D$	MV-DASE
2	$\neg f_B, \neg(f_G, f_D)$	$\neg f_B$, $\neg f_D$, $\neg f_G$, $\neg(f_G, f_D)$
3	BM25, avg, concat, ADA, word-level CCA, ELMo, $\neg f_U$, f_D, f_G, f_B, $\neg(f_B, f_U)$, f_U	BM25, avg, concat, doc2vec, ADA, word-level CCA, ELMo, f_B, f_D, f_G, $\neg f_U$, f_U, $\neg(f_B, f_U)$
4	InferSent, doc2vec	InferSent

Table 5: Group rankings by transitive closure of paired t-tests. $\neg f_j$ is MV-DASE without f_j (see Table 6). No particular order inside groups.

ADA. We evaluate the supervised domain-adversarial method of Shah et al. (2018) (ADA) on the low-resource forums. Recall that ADA requires a related labeled source domain. To achieve this, we pair every low-resource forum (target) with the CQADupStack forum (source) with which it has the highest word trigram overlap. See supplementary material for more details and a table of all source-target mappings.[14]

4.4 Ablation studies

We perform a set of experiments where we omit views from the ensemble. We also replace GCCA with naive view concatenation or view averaging. When averaging, we pad lower-dimensional vectors (Coates and Bollegala, 2018).

4.5 Significance tests

We perform paired t-tests, using the 20 test set forums as data points.[15] We then find groups of equivalent methods by transitive closure of $a \sim b \equiv p \geq .05$. Group A being ranked higher than group B means that every method in A performs significantly better than every method in B. Two methods in the same group may differ significantly, but there exists a chain between them of methods with insignificant differences.

5 Discussion

5.1 Comparison with baselines

BM25. BM25 is a tough baseline for DQD: In terms of MAP, it is better than or comparable to every single view (see Table 5). MV-DASE on the other hand, which is built from the same views, outperforms BM25 significantly and almost consistently (19 out of 20 test forums), regardless of the metric. This underlines the usefulness of our multi-view approach.

			MAP	AUC(.05)	NDCG	P@3	R@3
CQADupStack	1	$\neg f_G$	.002	.000	.000	.000	-.001
	2	$\neg f_D$	-.002	-.008	-.004	-.001	-.002
	3	$\neg f_U$	-.030	-.032	-.037	-.012	-.035
	4	$\neg f_B$	-.010	-.006	-.011	-.003	-.008
	5	$\neg(f_U, f_B)$	-.056	-.042	-.066	-.022	-.065
	6	$\neg(f_G, f_D)$	-.012	-.016	-.017	-.005	-.015
	7	concat	-.042	-.071	-.058	-.017	-.047
	8	avg	-.046	-.075	-.064	-.018	-.051
low-resource	9	$\neg f_G$	-.008	-.005	-.010	-.006	-.016
	10	$\neg f_D$	.007	-.001	.012	.002	.006
	11	$\neg f_U$	-.058	-.050	-.063	-.023	-.067
	12	$\neg f_B$	-.017	-.006	-.014	-.010	-.028
	13	$\neg(f_U, f_B)$	-.120	-.069	-.130	-.054	-.160
	14	$\neg(f_G, f_D)$	-.010	-.005	-.007	-.005	-.014
	15	concat	-.058	-.078	-.070	-.023	-.069
	16	avg	-.067	-.081	-.080	-.028	-.082

Table 6: Ablation study. Deltas relative to MV-DASE. Metrics were averaged over test forums before calculating deltas. concat/avg are naive view concatenation and averaging. Gray: better than MV-DASE.

Single views. MV-DASE outperforms the views that make up its ensemble significantly and almost consistently. There are two exceptions (out of 20 test forums): On *law* and *outdoors*, f_U (USE) performs slightly better on its own (Table 4, row 15). Since these forums are less "technical" than most, we hypothesize that they may be less in need of domain adaptation.

Word-level CCA. The word-level CCA baseline by Sarma et al. (2018) outperforms f_G and f_D on their own (see Table 4, rows 10, 21), which validates the approach. The method is directly comparable to MV-DASE$\neg(f_U, f_B)$, i.e., MV-DASE on generic and domain-specific averaged word embeddings (see Table 6). The main differences between them are (a) the order in which CCA and averaging are performed and (b) whether the CCA "vocabulary" is composed of word types or sentences. Note that in contrast to MV-DASE, word-level CCA is incompatible with contextualized embeddings, since it requires a context-independent one-to-one mapping between word types and vectors.

[14] We also experiment with Multinomial Adversarial Networks (MAN) (Chen and Cardie, 2018), a multi-source multi-target framework that can be trained on all 24 forums jointly. Initial results were not competitive with ADA, so we do not include them here. See supplementary material for details.

[15] Ten forums for t-tests involving ADA.

ADA. Supervised domain-adversarial ADA performs significantly worse than unsupervised MV-DASE (see Table 5). It is comparable to BM25 in terms of AUC(.05) (the metric used by Shah et al. (2018)), but not in terms of MAP.

Recall that we restricted the choice of source domains to the 12 CQADupStack forums. As a consequence, some target forums were paired with non-ideal source forums (e.g., *english→buddhism*). It is possible that the baseline would have performed better with a wider choice of source domains. Nonetheless, this observation highlights a key advantage of our approach: It does not depend on the availability of a label-rich related source domain (or indeed, any labels at all).

Other baselines. InferSent performs poorly on the DQD task, which is surprising given its similarity to USE. Recall that InferSent and USE are pre-trained on sentence-level SNLI, but that the training regime of USE also contains conversation response prediction. So USE is expected to be better equipped to handle (a) multi-sentence documents and (b) forum-style language.

Doc2vec is trained on the same data as f_D, but performs significantly worse. The difference between them may be due to the ability of FastText to exploit orthography. Domain-finetuned ELMo performs comparably to domain-finetuned BERT on some forums but not consistently.

5.2 Ablation study

View ablation. On the low-resource forums, omitting f_D has a beneficial effect (Table 6, row 10). This suggests that the in-domain FastText embeddings have insufficient quality when learned on the smallest forums and / or that domain-finetuned BERT subsumes any positive effect. On the high-resource CQADupStack forums, domain-specific embeddings contribute positively, while generic GloVe does not (rows 1,2). Table 5 shows that omitting either f_G or f_D from the ensemble does not lead to a significant drop in MAP, but omitting both does.

USE has the biggest positive effect on MV-DASE (Table 6, rows 3,11), also evidenced by the fact that omitting it is significantly more harmful than omitting any other single view (Table 5). Recall from Section 3.2 that USE is trained on supervised transfer tasks, while the remaining encoders are fully unsupervised.

GCCA ablation. The naive concatenation or averaging of views is significantly less effective than view correlation by GCCA (Table 6, rows 7,8,15,16, and Table 5). This underlines that multi-view learning is not just about *which* views are combined, but also about *how*. Intuitively, GCCA discovers which features from the different encoders "mean the same thing" in the domain. By contrast, concatenation treats views as orthogonal, while averaging mixes them in an unstructured way.

6 Evaluation on SemEval-2017 3B

In this section, we evaluate MV-DASE on SemEval-2017 3B, a DQD shared task based on the QatarLiving CQA forum. The benchmark provides manually labeled question pairs for training as well as additional unlabeled in-domain data. Since MV-DASE is unsupervised, we discard all training labels and concatenate training and unlabeled data into a text corpus ($\approx$ 1.5M tokens). This corpus is used for FastText training, BERT domain-finetuning, SIF weight estimation and GCCA, as described in Section 3.

The test set contains 88 queries q with ten candidates $c_1 \dots c_{10}$ each. We preprocess all data with the CQADupStack package and concatenate question subjects and bodies, before encoding them. We rank candidates by $\cos(f(q), f(c))$ and evaluate the result with the official shared task scorer.[16] In keeping with the original leaderboard, we report MAP and MRR (Mean Reciprocal Rank). We compare against previous literature as well as all individual views, view concatenation and averaging. See Table 7 for results. Like we observed on the Stack Exchange data, MV-DASE outperforms its individual views, their concatenation and average. It beats the previous State of the Art (a supervised system) by a margin of 2.5% MAP.

7 Evaluation on unsupervised STS

While this paper focuses on Duplicate Question Detection, MV-DASE is also applicable to other unsupervised sentence-pair tasks. As proof of concept, we test it on the unsupervised STS Benchmark (Cer et al., 2017). Here, the task is to predict similarity scores $y \in \mathbb{R}$ for sentence pairs (s_1, s_2).

[16] alt.qcri.org/semeval2017/task3/data/uploads/semeval2017_task3_submissions_and_scores.zip

		MAP	MRR
1	f_G (GloVe)	43.13	47.39
2	f_D (FastText)	43.38	47.67
3	f_U (USE)	48.22	52.73
4	f_B (BERT)	43.51	48.52
5	MV-DASE	51.56	56.48
6	concat	44.66	49.84
7	avg	44.95	49.76
8	Filice et al. (2017)*	49.00	52.41
9	Charlet and Damnati (2017)*	47.87	50.97
10	Goyal (2017)*	47.20	53.22
11	Zhang and Wu (2018)	48.53	52.75
12	Yang et al. (2018)	48.97	-
13	Gonzalez et al. (2018)	48.56	52.41
14	IR baseline*	41.85	46.42
15	Random baseline*	29.81	33.02

Table 7: MAP and MRR (percentages) on SemEval-2017 3B test set. *Shared task top teams (best run out of three) and baselines as reported in Nakov et al. (2017), Table 6. Gray: best in column.

		f_G = GloVe	f_G = ParaNMT
1	f_G	.731 / .647	.817 / .799
2	f_U (USE)	.793 / .762	.793 / .762
3	f_B (BERT)	.779 / .718	.779 / .718
4	MV-DASE	.825 / .771	.842 / .804
5	concat	.791 / .730	.826 / .772
6	avg	.790 / .729	.823 / .771

Table 8: Pearson's r (dev / test) on the unsupervised STS Benchmark, using different embeddings for f_G. Gray: best in column. Underlined: current unsupervised SoTA on test set (Wieting and Gimpel, 2018).

We treat the benchmark training set as an unlabeled corpus, i.e., we discard all labels and destroy the original sentence pairings by shuffling. The resulting corpus is used for BERT domain-finetuning, SIF weight estimation and GCCA. At test time, we measure Pearson's r between $\cos(f(s_1), f(s_2))$ and y, where f is an encoder (e.g., MV-DASE) and y is the ground truth similarity of test set pair (s_1, s_2).

In this experiment, the ensemble contains USE (f_U), domain-finetuned BERT (f_B) and f_G. For f_G, we either use SIF-weighted averaged GloVe vectors (Section 3.2), or unweighted averaged ParaNMT[17] word and trigram vectors (Wieting and Gimpel, 2018), which are the current State of the Art on the unsupervised STS Benchmark test set (Ethayarajh, 2018). The unlabeled training set is very small (64K tokens); hence, we do not include f_D in the ensemble, and we finetune the BERT language model for 10K rather than 100K steps to avoid overfitting. Like on the DQD tasks, MV-DASE beats its individual views as well as naive view concatenation and averaging (see Table 8). After adding ParaNMT to the ensemble, we achieve competitive results.

8 Future Work

Non-Linear GCCA. In Section 3.1, we assumed that relationships between representations are linear. This is probably reasonable for word embeddings (most cross-lingual word embeddings are linear projections, e.g. Artetxe et al. (2018)), but it is unclear whether it holds for sentence embeddings. Potential avenues for non-linear GCCA include Kernel GCCA (Tenenhaus et al., 2015) and Deep GCCA (Benton et al., 2017).

More views. A major advantage of MV-DASE is that it is agnostic to the number and specifics of its views. We plan to investigate whether additional or different views (e.g., encoders learned on related domains) can increase performance.

9 Conclusion

We have presented a multi-view approach to unsupervised Duplicate Question Detection in low-resource, domain-specific Community Question Answering forums. MV-DASE is a multi-view sentence embedding framework based on Generalized Canonical Correlation Analysis. It combines domain-specific and generic weighted averaged word embeddings with domain-finetuned BERT and the Universal Sentence Encoder, using unlabeled in-domain data only.

Experiments on the CQADupStack corpus and additional low-resource forums show significant improvements over BM25 and all single-view baselines. MV-DASE sets a new State of the Art on a recent DQD shared task (SemEval-2017 3B), with a 2.5% MAP improvement over the best supervised system. Finally, an experiment on the STS Benchmark suggests that MV-DASE has potential on other unsupervised sentence-pair tasks.

Acknowledgements

We thank Bernt Andrassy and Pankaj Gupta at Siemens MIC-DE, as well as our anonymous reviewers, for their helpful comments. This research was funded by Siemens AG.

[17] github.com/jwieting/para-nmt-50m

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*, Toulon, France.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*, pages 789–798, Melbourne, Australia.

Francis R Bach and Michael I Jordan. 2002. Kernel independent component analysis. *JMLR*, 3:1–48.

Timothy Baldwin, Huizhi Liang, Bahar Salehi, Doris Hoogeveen, Yitong Li, and Long Duong. 2016. UniMelb at SemEval-2016 Task 3: Identifying similar questions by combining a CNN with string similarity measures. In *International Workshop on Semantic Evaluation*, pages 851–856, San Diego, USA.

Adrian Benton, Huda Khayrallah, Biman Gujral, Dee Ann Reisinger, Sheng Zhang, and Raman Arora. 2017. Deep generalized canonical correlation analysis. *arXiv preprint arXiv:1702.02519*.

Dasha Bogdanova, Cicero dos Santos, Luciano Barbosa, and Bianca Zadrozny. 2015. Detecting semantically equivalent questions in online user forums. In *CoNLL*, pages 123–131, Beijing, China.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Danushka Bollegala and Cong Bao. 2018. Learning word meta-embeddings by autoencoding. In *COLING*, pages 1650–1661, Santa Fe, USA.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. In *ACL*, pages 730–740, Beijing, China.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642, Lisbon, Portugal.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *International Workshop on Semantic Evaluation*, pages 1–14, Vancouver, Canada.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *EMNLP*, pages 169–174, Brussels, Belgium.

Delphine Charlet and Geraldine Damnati. 2017. SimBow at SemEval-2017 Task 3: Soft-cosine semantic similarity between questions for community question answering. In *International Workshop on Semantic Evaluation*, pages 315–319, Vancouver, Canada.

Xilun Chen and Claire Cardie. 2018. Multinomial adversarial networks for multi-domain text classification. In *NAACL-HLT*, pages 1226–1240, New Orleans, USA.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. In *NAACL-HLT*, pages 194–198, New Orleans, USA.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, pages 670–680, Copenhagen, Denmark.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, Minneapolis, USA.

Cicero Dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. 2015. Learning hybrid representations to retrieve semantically equivalent questions. In *ACL*, pages 694–699, Beijing, China.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Workshop on Representation Learning for NLP*, pages 91–100, Melbourne, Australia.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*, pages 462–471, Gothenburg, Sweden.

Simone Filice, Giovanni Da San Martino, Alessandro Moschitti, and Roberto Basili. 2017. KeLP at SemEval-2017 Task 3: Learning pairwise patterns in community question answering. In *International Workshop on Semantic Evaluation*, pages 326–333, Vancouver, Canada.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *JMLR*, 17:2096–2030.

Ana Gonzalez, Isabelle Augenstein, and Anders Søgaard. 2018. A strong baseline for question relevancy ranking. In *EMNLP*, pages 4810–4815, Brussels, Belgium.

Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. O'Reilly Media.

Naman Goyal. 2017. LearningToQuestion at semeval 2017 Task 3: Ranking similar questions by learning to rank using rich features. In *International Workshop on Semantic Evaluation*, pages 326–333, Vancouver, Canada.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

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

Doris Hoogeveen, Andrew Bennett, Yitong Li, Karin M Verspoor, and Timothy Baldwin. 2018. Detecting misflagged duplicate questions in community question-answering archives. In *ICWSM*, Stanford, USA.

Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. 2015. CQADupStack: A benchmark data set for community question-answering research. In *Australasian Document Computing Symposium*, Parramatta, Australia.

Doris Hoogeveen, Karin M Verspoor, and Timothy Baldwin. 2016. CQADupStack: Gold or silver. In *Workshop on Web Question Answering Beyond Factoids*, volume 16, Pisa, Italy.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*, pages 328–339, Melbourne, Australia.

Jon R Kettenring. 1971. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *EMNLP*, pages 1466–1477, Brussels, Belgium.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NeurIPS*, pages 3294–3302, Montreal, Canada.

Jey Hand Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, Beijing, China.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*, pages 1073–1094, Minneapolis, USA.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *ICLR*, Vancouver, Canada.

Preslav Nakov, Doris Hoogeveen, Lluıs Marquez, Allessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community question answering. In *International Workshop on Semantic Evaluation*, pages 27–48, Vancouver, Canada.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, Doha, Qatar.

Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237, New Orleans, USA.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview LSA: Representation learning via generalized CCA. In *NAACL-HLT*, pages 556–566, Denver, USA.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Text REtrieval Conference (TREC)*, pages 109–126.

João António Rodrigues, Chakaveh Saedi, Vladislav Maraev, Joao Silva, and António Branco. 2017. Ways of asking and replying in duplicate question detection. In *Joint Conference on Lexical and Computational Semantics*, pages 262–270, Vancouver, Canada.

Prathusha K Sarma, Yingyu Liang, and William A Sethares. 2018. Domain adapted word embeddings for improved sentiment classification. In *Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 51–59, Melbourne, Australia.

Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *EMNLP*, pages 1056–1063, Brussels, Belgium.

Shuai Tang and Virginia R de Sa. 2019. Improving sentence representations with multi-view frameworks. In *Interpretability and Robustness for Audio, Speech and Language Workshop*, Montreal, Canada.

Arthur Tenenhaus, Cathy Philippe, and Vincent Frouin. 2015. Kernel generalized canonical correlation analysis. *Computational Statistics & Data Analysis*, 90:114–131.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008, Long Beach, USA.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *ACL*, pages 451–462, Melbourne, Australia.

Wei Yang, Wei Lu, and Vincent W Zheng. 2017. A simple regularization-based algorithm for learning cross-domain word embeddings. In *EMNLP*, pages 2898–2904, Copenhagen, Denmark.

Ziyi Yang, Chenguang Zhu, and Weizhu Chen. 2018. Zero-training sentence embedding via orthogonal basis. *arXiv preprint arXiv:1810.00438*.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *ACL*, pages 1351–1360, Berlin, Germany.

Mingua Zhang and Yunfang Wu. 2018. An unsupervised model with attention autoencoders for question retrieval. In *AAAI*, pages 4978–4986, New Orleans, USA.

Wei Emma Zhang, Quan Z Sheng, Jey Han Lau, and Ermyas Abebe. 2017. Detecting duplicate posts in programming QA communities via latent semantics and association rules. In *WWW*, pages 1221–1229.