Fusing Domain-Specific Data with General Data for In-Domain Applications

An-Zi Yen
National Taiwan University
Department of Computer Science and Information Engineering
Taipei, Taiwan
azyen@nlg.csie.ntu.edu.tw

Hen-Hsen Huang
National Taiwan University
Department of Computer Science and Information Engineering
Taipei, Taiwan
hhhuang@nlg.csie.ntu.edu.tw

Hsin-Hsi Chen
National Taiwan University
Department of Computer Science and Information Engineering
Taipei, Taiwan
hhchen@ntu.edu.tw

ABSTRACT

This paper analyzes the lexical semantics of domain-specific terms based on various pre-trained specific domain and general domain word vectors, and addresses the semantic drift between domains. To capture lexical semantics in the specific domain, we propose a bridge mechanism to introduce domain-specific data into general data, and re-train word vectors. We find that even a small-scale fusion can result in the similar lexical semantics learned by using the large-scale domain-specific dataset. Experiments on sentiment analysis and outlier detection show that application of word embedding by the fusion dataset has the better performance than applications of word embeddings by pure large domain-specific and pure large general datasets. The simple, but effective methodology facilitates the domain adaptation of distributed word representations.

CCS CONCEPTS

• **Information systems** → Information retrieval → Document representation → Content analysis and feature selection

KEYWORDS

Cross-Domain Data Fusion, Outlier Detection, Sentiment Analysis

ACM Reference format:

An-Zi Yen, Hen-Hsen Huang, Hsin-His Chen. 2017. Fusing Domain-Specific Data with General Data for In-Domain Applications. In *Proceedings of WI '17, Leipzig, Germany, August 23-26, 2017*, 7 pages.
http://dx.doi.org/10.1145/3106426.3106473

WI '17, August 23-26, 2017, Leipzig, Germany

ACM ISBN 978-1-4503-4951-2/17/08…$15.00
http://dx.doi.org/10.1145/3106426.3106473

1 INTRODUCTION

Word vectors used to capture lexical term semantics are crucial for many natural language tasks. Recently, word2vec [10] has been widely applied to construct such word vectors, and the resulting models lead to good performance in many tasks [3]. Nowadays pre-trained word vectors trained on large-scale datasets are available for researchers.

Word vectors learned from large-scale data may not always represent suitable semantics for target domains. For example, "reception" in hotel reviews represents "receptionist'". The related term is "front desk". By contrast, "reception" represents "ceremony" and "banquet" in other domain. In this paper, we deal with the semantic drift of domain-specific terms. We investigate word vectors trained on general and domain-specific datasets, and propose three mechanisms to introduce domain-specific data to general data. Word vectors derived from pure general dataset, pure domain-specific dataset, and fused dataset are analyzed and applied to sentiment analysis task and outlier detection task.

The major contribution of this work is threefold. (1) We propose a simple, but effective bridge mechanism to fuse domain-specific and general datasets. (2) We analyze the lexical semantics of domain-specific terms based on word vectors learned from domain-specific, general, and fused datasets in deep. (3) Applications on sentiment analysis task and outlier detection task show the feasibility of our methodology.

The rest of this paper is organized as follows. Section 2 introduces previous works on word embedding and domain adaption. Section 3 describes the proposed fusion mechanisms in details. Section 4 shows their applications on sentiment analysis. Section 5 discusses the results by analyzing word vectors trained on different datasets. Section 6 evaluates the performance of the methods on outlier detection task. Section 7 concludes the remarks.

2 RELATED WORK

Word embeddings have been shown effective in a variety of NLP tasks, such as tagging, chunking [3], parsing [1][13], machine translation [14][16][18] and sentiment analysis [8][15]. However, word embeddings learned on general datasets may not always capture the desired semantics of domain-specific terms for in-domain applications.

Different representation between source and target domains can be a major source of errors in domain adaptation. The task of domain adaptation aims at learning algorithms that can transfer from one domain to another, e.g., from social media to patient medical records.

Glorot et al. [7] propose a two-step procedure to deal with domain adaptation in sentiment analysis. They learn feature extraction using Stacked Denoising Autoencoder with sparse rectifier units, and then train a linear classifier with transformed labeled data of the source domain. Tang et al. [15] learn sentiment-specific word embedding. They develop three neural networks to effectively encode the supervision from sentiment polarity of text in the loss functions and map each n-gram to the sentiment polarity of sentence. Nguyen and Grishman [11] combine word clusters and word embedding information for domain adaption in relation extraction. While word clusters can be recognized as a one-hot vector representation, word embeddings are distributed representations. Kim [8] uses convolutional neural networks (CNN) to learn the task-specific vectors through fine-tuning for sentence-level classification tasks. The CNN model trains with two sets of word vectors, but gradients are backpropagated only through one of word vectors to fine-tune the vectors while keeping the other static and make it more specific to the task. Ding et al. [4] develop a Consumption Intention Mining Model (CIMM) which introduces a domain adaptation layer to convolutional neural network for identifying whether the user has a consumption intention. The Convolutional Layer can be viewed as feature extraction based on filters and it is able to capture the contextual features for a word. An adaptation layer transfers mid-level sentence representation from the source domain to the target domain. Yang and Eisenstein [17] focus on generating new feature representations for pivot features and propose FEMA (Feature EMbeddings for domain Adaptation) to learn low dimensional embeddings of the features for a CRF model by a variant of the Skip-gram Model [10] and achieves the state-of-the-art results on POS tagging adaptation tasks.

In this paper, we propose methods to modify the word representation of domain-specific and general domain words. We fuse domain-specific data into general data without changing the original word2vec learning model and analyze the lexical semantics of domain-specific terms. The word representation generated by our methods can be utilized in various classifiers like neural networks.

3 FUSION APPROACHES

In this study, the ClueWeb09-a dataset collected by CMU is considered as a general dataset. We extract 10 million sentences from this dataset as a general dataset, and collect 1,444,723 hotel reviews on TripAdvisor2 [5][9][12] as a domain-specific dataset. The reviews are rated by 1-5 stars. Only 1,011,339 reviews contain rating information. The reviews with star rating ≥3 are labeled as positive and the rest are negative in sentiment analysis [6]. There are 742,702 reviews labeled as positive and 268,637 reviews labeled as negative. The accuracy would be 73.44% for a classifier that always predicted the majority class. We compute term frequency (TF) and inverse document frequency (IDF) of words in the hotel review dataset, and use the TF-IDF weights to select the domain-specific terms.

As our methods are training word representations with Skip-gram model [9], we briefly introduce the Skip-gram algorithm. Its training objective is to learn word vector representations by predicting its context in the same sentence.

Given a context words $w_1, w_2, \ldots, w_T$, the objective of the Skip-gram model is to estimate the log probability of context words to be in the context of pivot word w_t:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} log\, p(w_{t+j}|w_t) \quad (1)$$

where T is the size of the training corpus, w_t is a target word, and c is the window size determining the span of context words of w_t. $p(w_{t+j}|w_t)$ is the probability of a context word given the target word.

According to the Skip-gram model, we propose the following methods to fuse domain-specific and general datasets.

Naïve Merge: Merge the domain-specific and the general datasets directly. In naïve merge, we do not distinguish the domain-specific terms in the datasets.

Restrictive Merge: We distinguish the domain-specific terms in domain-specific and general datasets. We label the top-N domain-specific terms t appearing in both datasets with domain labels s and g, i.e., t_s and t_g, which denote t's domain-specific and general uses. Assume "hotel" is a domain-specific term. We replace all "hotel" occurrences in the general dataset with a new word "hotel$_g$". Similarly, we substitute "hotel$_s$" for all "hotel" occurrences in the domain-specific dataset. After replacement, the two revised datasets are merged together. In this manner, the same domain-specific terms t in the two datasets are renamed. We can compare their domain-specific and general uses in the same embedding space. The remaining words without any domain labels are used to relate words in the domain-specific and general datasets.

Bridge Merge: We revise the Skip-gram model by the objective function as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\left(\left[\sum_{-c\le j<c, j\neq 0} log\, p(w_{t+j}|w_t)\right] + k(w_t)\right) \quad (2)$$

$$k(w_t) = \begin{cases} 0, if\ w_t\ is\ not\ a\ domain-specific\ term \\ log\, p(w_{t_s}|w_t), if\ w_t\ is\ a\ domain-specific\ term \\ \quad and\ \ in\ domain-specific\ dataset \\ log\, p\left(w_{t_g}|w_t\right), if\ w_t\ is\ a\ domain-specific\ term \\ \quad and\ in\ general\ dataset \end{cases}$$

[2] http://www.cs.cmu.edu/~jiweil/html/hotel-review.html

In the objective function of Bridge Merge, we add two additional probabilities $p(w_{t_s}|w_t)$ and $p\left(w_{t_g}|w_t\right)$ for the domain-specific term wt which appears in domain-specific and general datasets, respectively. Consider a hotel review in the domain-specific dataset: "This was a nice hotel". Under Restrictive Merge, this review will become "This was a nice hotel_s." In contrast, the original word, e.g., "hotel", acts as a bridge word between "hotel_g" and "hotel_s" in the Bridge Merge. In other words, domain-specific and general uses of a domain term share some information.

4 SENTIMENT ANALYSIS APPLICATION

Table 1 shows the datasets used to compute word embeddings. Besides pre-trained vectors trained on Google News dataset (G), we train word vectors with word2vec on the ClueWeb09-a dataset (C), the TripAdvisor dataset (T), and the three fused datasets created by Naïve Merge (N), Restrictive Merge (R), and Bridge Merge (B), where Skip-gram with dimension 300 and context size 5 are adopted.

We fuse different amounts A of data sampled from the TripAdvisor dataset into the ClueWeb09-a dataset, and create a fused dataset of size 170M+A. We compare the effects of different A in the experiments.

Table 1: Statistics of experimental datasets

Dataset	Abbr.	General/Specific	Size
Google News	G	General domain	100B
ClueWeb09-a	C	General domain	170M
TripAdvisor	T	Specific Domain	230M
Naïve Merge	N	Fused	170M+A
Restrictive Merge	R	Fused	170M+A
Bridge Merge	B	Fused	170M+A

We adopt 3-fold cross-validation to evaluate the accuracies of the sentiment analysis under different datasets and report the average accuracies in Table 2.

We compare the effects of different sizes of the domain-specific data in the experiments. Moreover, the up-pointing triangle (▲), the star (★), and the solid circle (●) denote the results are significant with p<0.001 using the McNemar's test comparing with Google News, ClueWeb09-a, and the TripAdvisor dataset, respectively.

We build each document embedding by summing up word vectors by equal weight as features, and classify reviews into positive review and negative review by linear SVM classifiers with L2-regularized L2-loss support vector classification dual and primal kernels to evaluate the effects of the word vectors trained on various datasets. The primal kernel is recommended when the number of instances is much larger than the number of features[3]. Table 2 lists the accuracies with three factors including (1) the adopted datasets, (2) the size of the domain-specific datasets, and (3) the size of general datasets.

Intuitively, classifier using the domain-specific dataset for in-domain applications is required. The experiments show classifiers using the domain-specific dataset (TripAdvisor) are better than those using the general datasets (GoogleNews and ClueWeb09-a) in both dual and primal kernels with significance level (p<0.001). Interestingly, classifiers fusing additional 3M-word domain-specific data are better than those using pure general datasets. The results show fusing domain-specific data is useful no matter which mechanism and which classification kernel are adopted. When dual kernel is used, Bridge Merge is better than Naïve Merge and Restrictive Merge. In some cases, Restrictive Merge is worse than Naïve Merge. When primal kernel is used, the difference among the three mechanisms is not distinct. However, Bridge Merge fusing additional 230M-word domain-specific data is 2.34% better than using the domain-specific dataset only (TripAdvisor). In addition, when the size of domain-specific data increases, the performance of Bridge Merge stably increases in both dual and primal kernels. Comparing the three merge methods, Bridge Merge is the best in both dual and primal kernels.

Table 2: Accuracies of different fusions in sentiment analysis application

Dataset	Specific domain size	General domain size	Accuracy (dual)	Accuracy (primal)
Google News	0	100B	86.54%	85.20%
ClueWeb09-a	0	170M	86.42%	85.42%
TripAdvisor	230M	0	87.71%	86.79%
Naïve Merge	3M	170M	86.81%▲★	87.13%▲★●
	15M	170M	87.23%▲★	87.82%▲★●
	38M	170M	87.66%▲★	88.62%▲★●
	77M	170M	87.79%▲★●	88.91%▲★●
	230M	170M	87.93%▲★●	89.04%▲★●
Restrictive Merge	3M	170M	87.02%▲★	85.66%▲★
	15M	170M	85.94%	87.81%▲★●
	38M	170M	86.76%▲★	88.55%▲★●
	77M	170M	87.04%▲★	88.84%▲★●
	230M	170M	87.05%▲★	88.87%▲★●
Bridge Merge	3M	170M	87.05%▲★	87.33%▲★●
	15M	170M	87.30%▲★	87.91%▲★●
	38M	170M	87.84%▲★●	88.48%▲★●
	77M	170M	87.95%▲★●	88.76%▲★●
	230M	170M	88.11%▲★●	89.13%▲★●

5 SEMANTICS OF DOMAIN-SPECIFIC TERM

To examine the effects of different datasets and merge strategies, we take 6 domain-specific terms as examples, where *check*, *distance*, *rate*, *reception*, and *staff* are aspect terms, and *helpful* is

[3] https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

an opinion word. Table 3 shows the top-6 similar terms of these 6 domain-specific terms. Here the size of the fused dataset is 173M words. To tell out the specific domain and the general domain semantics of the 6 observed terms in the fused datasets, we give a label *s* and *g* after the observed terms. The domain-specific interpretation is in bold.

Table 3: Top-6 similar terms of six in-domain terms in general, domain-specific, and fused datasets

D	word	top-1	top-2	top-3	top-4	top-5	top-6
G	check	recheck	doublecheck	verify	paystub	log	inspect
C	check	citybank	double-check	gemmologist	turn-wise	toondulge	post-dated
T	**check**	**checkin**	**checkout**	**smooth**	**painless**	**process**	**registration**
N	check	itaggit	lovefilms	**checkin**	double-check	re-check	buttom
R	check$_g$	doublecheck	reputation	citybank	verify	visit	advise
R	**check$_s$**	**checked$_s$**	**front$_s$**	**room$_s$**	**checkin$_s$**	**checkout$_s$**	**smooth$_s$**
B	check$_g$	checking$_g$	itaggit	myangelsname	back	visit	verify
B	**check$_s$**	**late$_s$**	**checkout$_s$**	**smooth$_s$**	**checkin**	**painless$_s$**	**process$_s$**
G	distance	withing_striking	shorter_distances	longdistance	mile	meters	twister_swirl
C	distance	miraflores	air-line	radius	**walking**	luxair	proximity
T	**distance**	**walking**	**walkable**	**distrance**	**attracitons**	**close**	**distence**
N	distance	**walking**	luxair	northbeach	air-line	miraflores	radius
R	distance$_g$	airline	miraflores	luxair	short$_g$	proximity$_g$	meters$_g$
R	**distance$_s$**	**walking$_s$**	**shopping$_s$**	**close$_s$**	**stroll$_s$**	**wandering$_s$**	**attractions$_s$**
B	distance$_g$	Luxair	computrain	trainingview	miles$_g$	long$_g$	short$_g$
B	**distance$_s$**	**close$_s$**	**walkable$_s$**	**walking$_s$**	**blocks$_s$**	**attracitons**	**reatsurants**
G	helpful	beneficial	invaluable	handy	informative	valuable	enlightening
C	helpful	informative	beneficial	knowledgeable	thoughful	valuable	confusing
T	**helpful**	**courteous**	**accommodating**	**friendly**	**polite**	**attentive**	**professional**
N	helpful	**courteous**	knowlegeable	**accommodating**	beneficial	muchly	advice
R	helpful$_g$	beneficial$_g$	informative$_g$	invaluable$_g$	valuable$_g$	interesting$_g$	difficult$_g$
R	**helpful$_s$**	**friendly$_s$**	**staff$_s$**	**concierge$_s$**	**polite$_s$**	**attentive$_s$**	**professional$_s$**
B	helpful$_g$	beneficial$_g$	beneficial	informative$_g$	valuable$_g$	found$_g$	polite$_g$
B	**helpful$_s$**	**friendly$_s$**	**staff$_s$**	**concierge$_s$**	**polite$_s$**	**gracious$_s$**	**obliging$_s$**
G	rate	interest_rates	uninfected_chimps	percentage	rediscount_rate	borrowing_costs	interestrate
C	rate	interest	blr	rba	mortgage	percentage	interest-rates
T	**rate**	**price**	**discounted**	**deal**	**promotional**	**expedia**	**travelocity**
N	rate	percentage	amortisation	Libor	interest-rates	pro-cyclical	real-wage
R	rate$_g$	percentage$_g$	tranfer	Sibor	calculated	mortgages	payments$_g$
R	**rate$_s$**	**price$_s$**	**deal$_s$**	**paid$_s$**	**pricing$_s$**	**bargain$_s$**	**discount$_s$**
B	rate$_g$	libor	interest$_g$	percentage$_g$	average	percent$_g$	calculated
B	**rate$_s$**	**price$_s$**	**paid$_s$**	**booked$_s$**	**discount$_s$**	**deal$_s$**	**competitive$_s$**
G	reception	ceremony	banquet	luncheon	accorded_rousing	beatific_funeral	dinner
C	reception	ceremony	unbelievably	dinner	deuvres	wedding	bridal
T	**reception**	**receptionist**	**front**	**desk**	**recetion**	**foyer**	**recepton**
N	reception	journeystm	**helfpul**	ceremony	post-wedding	**manager**	celebration
R	reception$_g$	ceremony$_g$	wedding$_g$	unbelievably$_g$	dinner$_g$	banquet$_g$	channel$_g$
R	**reception$_s$**	**desk$_s$**	**staff$_s$**	**lobby$_s$**	**receptionist$_s$**	**foyer$_s$**	**concierge$_s$**
B	reception$_g$	journeystm	ceremony$_g$	wedding$_g$	banquet$_g$	evening$_g$	rehearsal$_g$
B	**reception$_s$**	**desk$_s$**	**front$_s$**	**lobby$_s$**	**foyer$_s$**	**concierge$_s$**	**manager$_s$**
G	staff	staffers	personnel	assistants	employees	interns	staffing
C	staff	faculty	personnel	employees	radiographers	volunteers	storea
T	**staff**	**friendly**	**polite**	**personnel**	**courteous**	**helpful**	**employees**
N	staff	**courteous**	**helpful**	**knowlegeable**	volunteers	baby-sitters	cross-trained
R	staff$_g$	personnel$_g$	faculty$_g$	employees$_g$	members	volunteers	systel
R	**staff$_s$**	**friendly$_s$**	**helpful$_s$**	**extremely$_s$**	**professional**	**welcoming**	**concierge$_s$**
B	staff$_g$	personnel$_g$	faculty$_g$	systel	volunteers	consultants$_g$	trained$_g$
B	**staff$_s$**	**friendly$_s$**	**helpful$_s$**	**hotel$_s$**	**professional$_s$**	**gracious$_s$**	**welcoming**

The top-3 similar terms in general datasets (G and C) are not related to the domain-specific terms. The word "check" in G and C represents the meaning of examination and banking, but it is related to check in, check out, or the behavior of checking in hotel reviews.

The similar words of the word "staff" in general domain G and C are the synonyms of employee, but the similar words in specific domain focus on attitudes of the hotel staffs. These examples demonstrate word vectors trained on huge datasets like Google News and ClueWeb09-a cannot fully capture lexical semantics of the domain-specific terms.

Besides, comparing Naïve merge and Bridge Merge, the semantics of the domain-specific terms in Naïve merge are more general than that in Bridge Merge, e.g., the word "rate" in Naïve merge is related to economy. In specific domain of Bridge Merge, by contrast, it is related to the price of hotel room. Therefore, the word representations learned from Bridge Merge is necessary for specific domain application.

We further analyze the differences between Restrictive Merge and Bridge Merge. We draw the scatter plots of the word vectors trained on datasets by these two methods. Principal Component Analysis (PCA) is used to reduce dimension. Fig. 1 and Fig. 2 show the results for Bridge Merge and Restrictive Merge, respectively. The blue point denotes t_s; the red point denotes t_g; and the yellow point denotes the other words. Fig. 2 shows t_s and t_g are separated, including some opinion words like "good" and "bad". However, the opinion words "good" and "bad" should represent similar semantics in general domain and specific domain. Our goal is to distinguish the slightly different semantics of the specific domain words from general domain. Fig. 2 shows Restrictive Merge separates the word representations too much. Comparatively, in Fig. 1, t_s are grouped together, and are surrounded by t_g with the other words as bridge. Besides, the specific domain words and general domain words still contain domain semantics. For example, "reception$_s$" and "reception$_g$" are not close to each other in the embedding space. The opinion words "good" and "bad" from both domains are close together, and separate from other specific domain words and aspect terms.

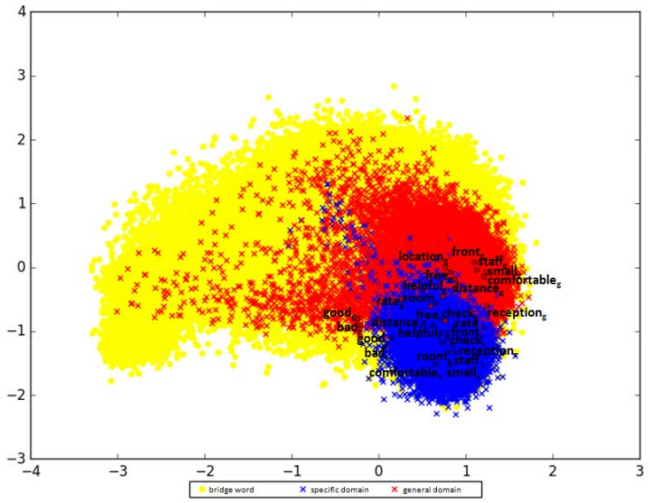

Figure 1: Plot for bridge merge.

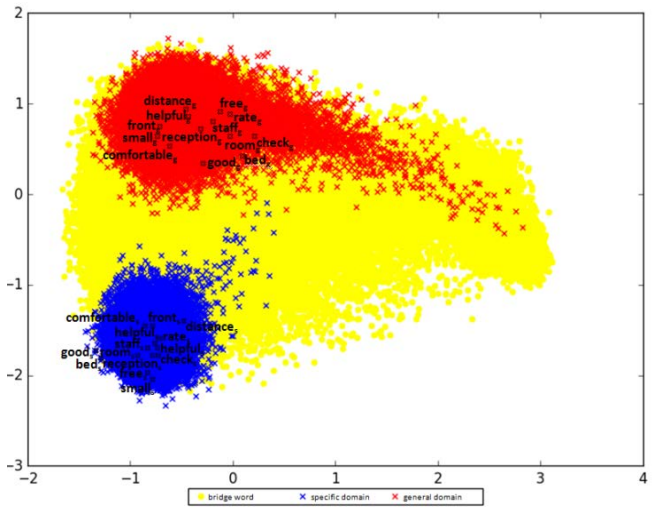

Figure 2: Plot for restrictive merge.

We also compute a similarity list of t_s and a similarity list of t_g based on the word representations generated by Bridge Merge and Restrictive Merge. Table 4 shows the ranks of t_s in t_g's similar term list are higher than those of t_g in t_s's similar term list. For example, the word *check* in domain-specific use ranks 258 in the similarity list of *check* in general use. Comparatively, the word *check* in general use ranks 717 in the similarity list of *check* in domain-specific use. That demonstrates the differences of domain-specific use and general use can be distinguished by the proposed Bridge Merge. In contrast, Table 5 shows both of the ranks of similar term lists in Restrictive Merge are low. It seems that there is no correlation between general and specific domains and the word representations of domain-specific terms do not effectively utilize the information of general domain to learn the semantic of specific domain.

Table 4: Similarity of t_s and t_g in bridge merge

t	rank of t_s in t_g list	rank of t_g in t_s list
check	258	717
distance	19	205
helpful	597	4,054
rate	10	62
reception	22	270
staff	1,652	9,490
location	918	6107

Table 5: Similarity of t_s and t_g in restrictive merge

t	rank of t_s in t_g list	rank of t_g in t_s list
check	43,078	31,091
distance	28,686	18,461
helpful	189,580	83,645
rate	96,617	28,375
reception	28,395	51,479
staff	279,826	130,317
location	278,614	146,305

6 OUTLIER DETECTION TASK

In this section, we aim at evaluating our fusing methods by an intrinsic evaluation of word vector representations. The proposed task is inspired from the GRE antonym detection task, whose goal is to identify the word not belonging to a given group. In contrast to the antonym detection task, in this paper, we evaluate a model by predicting which word does not belong to the category of the remaining words. First of all, we define six categories which are related to the hotel review. The six categories are *room*, *service*, *meal*, *location*, *cost*, and *facility*. Then, we extract the top 100 aspect terms ranked by TF-IDF weights and label each word with one category, e.g., breakfast, buffet, and fruit are labeled as meal. Finally, we list all kinds of questions which contain three aspect words in the same category and one word belonging to the other category. There are 372,531 questions in this task.

Table 6: Accuracies of different fusions in outlier detection application

Dataset	Specific domain size	General domain size	Accuracy
Google News	0	100B	54.60%
ClueWeb09-a	0	170M	53.28%
TripAdvisor	230M	0	66.50%
Naïve Merge	3M	170M	52.84%
	15M	170M	56.87%▲★
	38M	170M	59.30%▲★
	77M	170M	58.88%▲★
	230M	170M	53.28%
Restrictive Merge	3M	170M	62.53%▲★
	15M	170M	65.02%▲★
	38M	170M	64.91%▲★
	77M	170M	63.65%▲★
	230M	170M	66.67%▲★●
Bridge Merge	3M	170M	59.28%▲★
	15M	170M	66.22%▲★
	38M	170M	66.90%▲★●
	77M	170M	66.88%▲★●
	230M	170M	67.66%▲★●

Consider an example question: 1. elevator, 2. rate, 3. expensive, 4. cheap. Here "elevator" belongs to the category of *facility* and the other words belong to the category of *cost*. For evaluating the performance of Restrictive Merge and Bridge Merge, we use the terms in domain-specific use like elevator, and rate. In each question, we choose the word with the smallest similarity score as an answer. The similarity score is calculated by averaging all pair-wise semantic similarities of the words in question [2]. Table 6 shows the prediction results measured by accuracy. McNemar's test is adopted for significance test ($p<0.001$). Meanings of the symbols ▲, ★, and ● are the same as those defined in Section 4.

We have the following findings. Intuitively, using the pure domain-specific dataset (TripAdvisor) outperforms using the general domain datasets (Google News and ClueWeb09-a). Naïve Merge and Restrictive Merge are also inferior to TripAdvisor except Restrictive Merge fusing all TripAdvisor data. Bridge Merge, the best fusing method, introducing only additional 38M-word domain-specific data significantly outperforms the pure domain-specific dataset.

7 CONCLUSIONS

This paper addresses the effects of different interpretations in domain-specific use and general use on various applications, and proposes a simple but effective mechanism to deal with this problem. The experiments on the applications of sentiment analysis and outlier detection show the word vectors trained on the dataset fusing domain-specific data with general dataset not only capture clear lexical semantics, but also have better accuracies than the Google pre-trained word vectors.

In the future, we will extend the methodology to learn the domain-specific terms semantics of other domains (e.g., medical, product, etc.). For example, in the camera reviews, the word big might not be a positive opinion word. "Very big to hold" is a negative opinion about the camera. Besides, the words sleek and lightweight implicitly provide a positive opinion about the aspects appearance and weight of the entity camera. In addition, personal health or medical data are rare and the amount of data is less. We can apply our methods to fuse domain-specific and general datasets to learn the word representations in specific domain more precisely.

Acknowledgments

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-104-2221-E-002-061-MY3 and MOST-105-2221-E-002-154-MY3.

REFERENCES

[1] D. Chen, and C. D. Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2014). ACL, 740–750.

[2] J. Camacho-Collados, and R. Navigli. 2016. Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP* (ACL 2016). ACL, 43–50.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa. 2011. Natural language processing (almost) from scratch. In *Journal of Machine*

Learning Research 12 (2011) 2493–2537.

[4] X. Ding, T. Liu, J. Duan and J. Y. Nie. 2015. Mining User Consumption Intention from Social Media Using Domain Adaptive Convolutional Neural Network. In *Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2389–2395.

[5] K. Ganesan, and C. Zhai. 2012. Opinion-Based Entity Ranking. *Information Retrieval 15, 2 (2012)*, 116–150. DOI=http://dx.doi.org/10.1007/s10791-011-9174-8

[6] G. Gezici, B. Yanikoglu, Dilek Tapucu, and Yücel Saygın. 2012. New features for Sentiment Analysis: Do Sentences Matter. In *Proceedings of The International Workshop on Sentiment Discovery from Affective data*. 5–15.

[7] X. Glorot, A. Bordes, and Y. Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classication: A Deep Learning Approach. In *Proceedings of the 28th International Conference on Machine Learning* (ICML-11). 513–520.

[8] Y. Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2014). ACL, 1746–1751.

[9] D. Marcheggiani, O. Täckström, A. Esuli, and F. Sebastiani. 2014. Hierarchical Multi-label Conditional Random Fields for Aspect-oriented Opinion Mining. In *Proceedings of the 36th European Conference on IR Research on Advances in Information Retrieval - Volume 8416 (ECIR 2014)*. 273-285. DOI=http://dx.doi.org/10.1007/978-3-319-06028-6_23

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceeding NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*. 3111–3119.

[11] T. H. Nguyen, and R. Grishman. 2014. Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (ACL 2014). ACL, 68–74.

[12] H. Wang, C. Wang, C. Zhai, and J. Han. 2011. Learning Online Discussion Structures by Conditional Random Fields. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11)*. ACM, New York, USA, 435-444. DOI= http://dx.doi.org/10.1145/2009916.2009976

[13] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*. ACL, 455–465.

[14] I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceeding NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*. 3104–3112.

[15] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. ACL, 1555–1565.

[16] I. Vulić, and M. F. Moens. 2015. Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL 2015)*. ACL, 719–725.

[17] Y. Yang, and J. Eisenstein. 2015. Unsupervised multi-domain adaptation with feature embeddings. In *Proceedings of the 2014 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*. 672–682.

[18] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning. 2013. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*. 1393–1398.