Distributed Word Representations Improve NER for e-Commerce

Mahesh Joshi
mahesh.joshi@ebay.com

Ethan Hart
ejhart@ebay.com

Mirko Vogel
miavogel@ebay.com

Jean-David Ruvini
Jean-David.Ruvini@ebay.com

eBay Inc.
2065 Hamilton Ave
San Jose, CA, 95125, USA

Abstract

This paper presents a case study of using distributed word representations, word2vec in particular, for improving performance of Named Entity Recognition for the e-Commerce domain. We also demonstrate that distributed word representations trained on a smaller amount of in-domain data are more effective than word vectors trained on very large amount of out-of-domain data, and that their combination gives the best results.

1 Introduction

On-line commerce has gained a lot of popularity over the past decade. Large on-line C2C marketplaces like eBay, Alibaba, and Amazon feature a very large and long-tail inventory with millions of *items* (product offers) entered into the marketplace every day by a large variety of sellers.

To manage items effectively and provide the best user experience, it is critical for these marketplaces to structure their inventory into descriptive name-value pairs (called *properties*) and ensure that items of the same kind (digital cameras, for instance) are described using a unique set of properties (brand name, model number, zoom, resolution, etc.). This is important for recommendations in merchandising, providing faceted navigation, and assisting business intelligence applications.

While some sellers (generally large, professional retailers) provide rich, structured descriptions of their products (using schemas or global trade item numbers), the vast majority of sellers only provide unstructured natural language descriptions. In the latter case, one solution to the problem of structuring e-commerce inventory is to use techniques such as Named-Entity Recognition (NER) to extract properties from the textual description of the items. The scale at which on-line marketplaces operate makes it impractical to solve this problem manually. [1]

This paper focuses on NER, generally defined as the task of classifying elements of text into predefined categories (often referred to as *entity types* or *entities*). Entities usually include names of persons, organizations, locations, times, and quantities (CoNLL-2003 dataset), as well as nationalities or religious groups, products (vehicles, weapons, foods, etc.), and titles of books or songs (Ontonotes 5.0 dataset).

In the e-commerce domain, these entities are item properties such as brand name, color, material, clothing size, golf club type, makeup shade code, sun protection factor, etc. Another important specificity of the e-commerce domain with respect to NER is that the sentences are usually much shorter than in other applications and don't exhibit the grammatical structure of natural language.

This paper investigates whether distributed word vectors benefit NER in the e-commerce domain. Distributed word representations based on neural networks from unlabeled text data have proven useful for many natural language tasks, including NER. In fact, Passos et al. (2014) reported results compa-

[1] For instance, in late 2014, eBay.com reported 800 million available items at any given time and more than 25 million sellers. Alibaba.com reported 8.5 million sellers. Amazon.com has not disclosed similar information.

Proceedings of NAACL-HLT 2015, pages 160–167,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

rable to state-of-the-art for the CoNLL 2003 NER task using such representations. In this paper, we evaluate distributed word vectors with a focus on using in–domain data for their training.

In the remainder of this paper, we first explain the specificity of NER in the e-commerce domain and describe the approach we use for performing the task. In Section 3, we describe our datasets. In Section 4, we describe the setting of the experiments we have conducted and discuss the results in Section 5. Finally, we review related works in Section 6.

2 NER for e-Commerce

The e-commerce domain raises specific challenges for NER. This section describes in detail the task, and the methodology we have chosen to tackle it.

2.1 Description of the task

We consider the task of named entity recognition (NER) on text from the e-commerce domain. The text data associated with an e-commerce item usually consists of two parts: the *title* and the *description*. In the current work, we focus only on item titles since item descriptions are often optional, vary greatly from seller to seller and between marketplaces, and are not shown on the search results page. The item title is a short sentence usually consisting of a sequence of approximately 10 to 35 nouns, adjectives, and numbers. They rarely contain verbs, pronouns, or determiners. The title is mandatory for most marketplaces, as it is indexed by the search engine and searched against by users of the website. Snippets shown in search result pages are generated from the titles of the items in the search result set.

Table 1 shows some examples of item titles (rows 1, 3, 5) from various online marketplaces. These examples show that sellers use capitalization and special characters as visual features in a manner not necessarily consistent with conventional English grammar rules. Besides their limited grammatical structure and the lack of contextual information due to their length, titles also contains typographical errors and abbreviations. While many abbreviations are standard in the e-commerce domain and used across all marketplaces (such as “w/” for “with”, “NIB” for “new in box”, “BNWT” for “brand new with tag”, etc.), some are seller specific and are often difficult to decipher.

Performing NER for e-commerce involves classifying the various tokens in the title of an item into property names (entities) relevant to that item. Table 1 also shows the annotated entities (rows 2, 4, 6) for each of the titles. Section 3 provide details about the e-commerce categories, and the empirically defined entities within each of those categories. Next, we describe the approach that we use for NER.

2.2 Approach

Following current best practices, we approach NER as a sequence labeling problem. We use linear–chain Conditional Random Field (CRF) (Lafferty et al., 2001) which has been shown to achieve the best performance for many applications of NER (Suzuki and Isozaki, 2008; Lin and Wu, 2009; Passos et al., 2014), including NER for the e-commerce domain (Putthividhya and Hu, 2011).

We use a fairly standard set of lexical features used in most NER systems, including character affixes. Our features are detailed in Section 4.

In addition to the lexical features, modern NER systems also attempt to leverage some form of vector representation of the syntactic and semantic properties of the tokens. While discrete word representations derived from word clusters have been shown to be very beneficial to NER (Miller et al., 2004; Lin and Wu, 2009; Ratinov and Roth, 2009; Turian et al., 2010), more and more attention is being paid to distributed word representations since the introduction of efficient algorithms to produce them (Mikolov et al., 2013). Passos et al. (2014), for instance, reported performance comparable to state-of-the-art NER systems using a modified skip–gram model trained to predict membership of words to a domain specific lexicon.

To the best of our knowledge, all the results reported so far for NER used distributed word vectors trained from documents composed in standard, mostly grammatical English (Collobert and Weston, 2008; Turian et al., 2010; Baroni et al., 2014; Passos et al., 2014). However, it is clear that some phrases in the e-commerce domain have a very different meaning than in conventional English. For instance, “adventure time,” “baby, the stars shine bright,” and “miss me” are a few examples of e-commerce brand names which occur rarely in Wikipedia. In

1:	Apple	iPhone	6	-	16GB	-	Space	Grey	(	Unlocked	)	Smartphone
2:	b	p	p		d		c	c		a		t
3:	Cole	Haan	Men	’s	Carter	Grand	Cap	Oxford				
4:	b	b	g	g	p	p		t				
5:	Womens	CRISTINALOVE	SHOULDER	DRESS	-	Size	XL	-	L@@K			
6:		b	t	t			s					

Table 1: Examples of annotated titles for different e-commerce categories from various online marketplaces. Entity types are denoted by the single letters: “a”– “contract,” “b”– “brand name,” “c”– “color,” “d”– “dimension,” “g”– “gender,” “p”– “product name or number,” “t”– “type,” “s”– “size.”

this paper, we investigate whether useful distributed representations can be learned from fairly unstructured, short, ungrammatical documents such as e-commerce titles and capture enough e-commerce semantics to benefit NER. We also study how they compare to distributed vectors trained from a non e-commerce corpus.

3 Data

To make discovering and browsing the inventory easy, most on-line marketplaces organize their inventory into a category structure similar to a topic hierarchy. eBay and Alibaba hierarchies comprise around 40 top level nodes, called *categories*, and more than 10,000 leaf nodes. The goods from different categories are usually very different in nature as exemplified by eBay categories such as “Antiques,” “Clothing, Shoes & Accessories,” and “Toys & Hobbies,” to name a few.

3.1 Data Selection

The models trained for our experiments focus on a subset of five popular categories, namely Cellphones (CELLPH), Cellphone Accessories (CELLACC), Men’s Shoes (MSHOES), Watches (WATCHES), and Women’s Clothing (WCLOTH). Our datasets consist of user-defined e-commerce item titles. Table 2 provides statistics about these titles. Titles were tokenized using CoreNLP (Manning et al., 2014).

3.2 Training and testing data

Training and testing data for CRF was produced by manually labeling data. Based on the labeling resources available, we sampled 2,000 titles for most categories. Splitting these samples resulted in the training and test splits shown in Table 3.

category	# titles	# tokens	vocab. size
CELLPH	29M	46M	23K
CELLACC	143M	1.8B	114K
MSHOES	61M	665M	95K
WATCHES	97M	959M	190K
WCLOTH	150M	1.6B	118K

Table 2: Approximate statistics for the in-domain titles (B: billion, M: million, K: thousand). The vocabulary size is based on a minimum count of 50.

category	titles	tokens	vocab
CELLPH	1500 / 500	20776 / 7056	3806 / 1647
CELLACC	1330 / 443	18650 / 6195	4964 / 2261
MSHOES	1485 / 494	19278 / 6373	5424 / 2513
WATCHES	1339 / 495	15735 / 5828	5176 / 2487
WCLOTH	3098 / 500	39196 / 6279	7576 / 2621

Table 3: Training / test data splits (titles, token count, vocabulary size) for each category.

3.3 Entity Types

An important step in preparing the data was determining which properties of the items are most important to each category (concretely, which entities should be targeted). Because items across categories are quite different and can vary greatly in nature, a unique set of entities was used for each category, though several entities are common across categories (e.g. `brand`, `color`). For example, a title in WCLOTH might contain the properties `brand`, `type`, `size`, `style`, `color` whereas a title in CELLPH might describe an item by `brand`, `product name`, `storage size`, `contract`. These tags were chosen based on frequently occurring, user-defined properties that are assigned to an item. This set was manually pared down based on how much coverage an entity set could achieve while maintaining a manageable number of entities. While it would be ideal to have a set of entities such that every word in a title is tagged, this does not scale well and makes the annotation task more difficult. Table 4 shows the set of entities used for each category and the distribution of entities over the individual tokens within each category.

3.4 Annotation Procedure

Titles in Table 3 were annotated by two language specialists. Annotators had access to the listing page of the item in question to use as a reference. This page typically includes some pictures as well as a description of the item which may provide information about a particular token and reduce the amount of research required to correctly label a token (e.g. an obscure brand name). The two annotators regrouped after tagging to resolve disagreements between the individually tagged data sets. Agreement scores between the annotators were calculated using unweighted Cohen's Kappa with the following results: CELLPH: .92, CELLACC: .82, MSHOES: .78, WATCHES: .81, WCLOTH: .93. BIO encoding was not used for these datasets, but experimenting with it is important, and we plan do so in future work.

3.5 `word2vec` training data

For training the category–specific in–domain word vector representations, the set of tokenized titles referred to in Table 2 are used for the respective category. Section 4 provides details about the `word2vec` training process.

4 Experiments

We now present our experimental results for NER on e-commerce item titles. The goal of our work is not necessarily to present the best possible results for this task. Instead, our experiments are driven by the following two questions: (1) Are distributed word representations created from highly unstructured data (namely, e-commerce item titles) beneficial for the task of named entity recognition on the same kind of unstructured data? (2) How do distributed word vector representations created from out-of-domain (namely, non e-commerce data) compare with those created from in-domain data?

4.1 Training

We use the CRFsuite package (Okazaki, 2007) for our experiments. Following Turian et al. (2010), we use stochastic gradient descent (SGD) for our training, and allow negative state features and negative transition features. The l_2 regularization hyper–parameter (`c2` for CRFsuite) is tuned using a randomly chosen subset of 30% sentences (item titles) held out as the development set during training. The final model is retrained on the entire training set using the best value of `c2` (which varies depending on the feature configuration). The set of `c2` values we tried is {`0.001, 0.005, 0.01, 0.05, 0.1, 1, 2, 5, 10, 50, 100`}.

feature	**comment**
$\mathtt{w}_0$, $\mathtt{w}_{-1}$, $\mathtt{w}_{+1}$	current token, tokens in window of 1
$\langle \mathtt{w}_{-2}, \mathtt{w}_{-1} \rangle$, $\langle \mathtt{w}_{+1}, \mathtt{w}_{+2} \rangle$	left and right bigram
CLASS($\mathtt{w}_{-1}$, $\mathtt{w}_0$, $\mathtt{w}_{+1}$)	`ALLCAPS`, `Initcap`, `UpperCamelCase`, etc.
$\lvert \mathtt{w}_0 \rvert$	length of current token
RELPOS($\mathtt{w}_0$)	relative position in the item title
AFFIXES($\mathtt{w}_{-1}$, $\mathtt{w}_0$, $\mathtt{w}_{+1}$)	up to 3-character prefixes and suffixes
$\mathtt{t}_{-1}$	tag of the previous token

Table 5: Table shows the features that we use for our baseline.

4.2 Baseline Features

Table 5 shows the features that we use for our baseline. We refer to this feature set by the name **BASE** in our results section.

We also experimented with larger window sizes (two and three) for all of the windowed features listed in Table 5, however, the performance degraded for larger window sizes. We believe this is due to the highly unstructured nature of text in the item titles.

4.3 Distributed Word Vector Features

We explored two different types of sources of text for the generation of distributed word representations for our task. First, we used word vectors trained by Baroni et al. (2014) — in particular, the "best predict vectors" made available by the authors[2]. These are, for our purposes, vectors trained on out-of-domain text corpora. Results using features based on these word vectors are denoted by the name **BASE+GEN**. In our experiments, features based on word vectors are always added on top of

[2] `http://clic.cimec.unitn.it/composes/semantic-vectors.html`

Category	Tag
CELLPH	Product Name 17.8%, Brand 11.3%, Dimension 7.4%, Color 5.2%, Contract 3.8%, Operating System 2.1%, Location 0.4%, no tag 52.0%
CELLACC	Product Name 25.6%, Type 19.4%, Brand 7.3%, Feature 3.8%, Material 2.4%, Color 2.3%, Style 1.2%, Connectivity 1.1%, Pattern 0.6%, Battery Capacity 0.4%, Location 0.4%, Finish 0.3%, Fit 0.3%, Storage Capacity 0.3%, Storage Format 0.2%, Sports Team 0.2%, no tag 34.2%
MSHOES	Product Line 28.2%, Brand 11.4%, Color 8.8%, Size 6.9%, Type 5.6%, Gender 3.9%, Purpose/Occasion/Activity 2.9%, Material 2.2%, Height 1.3%, Style 1.1%, Pattern 0.4%, no tag 27.3%
WATCHES	Product Name 13.8%, Brand 9.3%, Type 7.4%, Feature 5.1%, Gender 4.7%, Material 4.0%, Color 3.1%, Movement 2.9%, Component 2.9%, Style/Purpose/Occasion 2.9%, Size 1.1%, Display Type 0.8%, Location 0.8%, Purity/Quality 0.3%, Shape 0.2%, Closure 0.1%, no tag 40.6%
WCLOTH	Type 16.0%, Brand 8.3%, Size 7.3%, Color 4.0%, Material 3.8%, Purpose/Occasion/Activity 3.2%, Style 2.1, Pattern 1.5%, Location 0.8%, no tag 53.0%

Table 4: The entities targeted by our NER system and their distributions over total tokens for each category.

our baseline features (**BASE**). Second, we used word vectors trained on a large set of in-domain data for each of the five categories, namely e-commerce item titles for the respective categories. The word vectors for each category were trained separately, in order to provide the "purest" form of in–domain data. Results using features based on these word vectors are denoted by the name **BASE+DOM**. Additionally, we also conduct experiments using features based on both the in–domain as well as out–of–domain word vectors. Results using this combined set of word vector features are denoted by **BASE+ALL**.

Word vector features are computed for $\mathtt{w}_0$, $\mathtt{w}_{-1}$, and $\mathtt{w}_{+1}$— that is, for the current token and its two surrounding tokens. Here too, we experimented with larger window sizes, but that resulted in a lower overall performance.

4.4 word2vec

Both the out–of–domain and the in–domain word vectors that we train are trained using the word2vec toolkit[3] (Mikolov et al., 2013). Details of how the out–of–domain word vectors were trained is provided by Baroni et al. (2014) — their 400–dimensional word vectors were trained on approximately 2.8 billion tokens using word2vec's continuous bag–of–words (cbow) representation, with a window size of five.

Initially, we experimented with several parameter choices for training our word2vec models. In particular, we tried the following grid of values: representation: skip-gram, continuous bag–of–words (cbow); context window size: {2, 5}, down–sampling parameter: {1e-3, 1e-4, 1e-5}; hierarchical softmax: {off, on}; # of negative samples: {5, 10}; word frequency cutoff: {10, 50}; and word vector dimensionality: {50, 100, 200, 300, 400, 500}. Based on this parameter sweep, we found that the following parameters worked best overall for our task: representation: skip-gram, context window size: 2, down–sampling parameter: 1e-3, hierarchical softmax: off, # negative samples: 10, word frequency cutoff: 50. These are the settings we use for all the results reported in this paper. As for the word vector dimensionality, we tuned it based on our validation set (similar to the c2 parameter for CRFsuite), using the following set of values: {50, 100, 200, 300, 400, 500}). In our results we will report the best word vector dimensionality when features based on in–domain word vectors are used.

The skip-gram representation worked better in our experiments for capturing semantics of the word co-occurrences in the item titles. This is consistent with the comparative analysis published by Mikolov et al. (2013) between skip-gram and cbow models — the cbow models were found to be better for syntactic tasks while the skip-gram models were better for semantic tasks. A narrower context window is better for our highly unstructured data.

5 Results and Discussion

Table 6 shows our complete set of results. We report the weighted token–level precision, recall, and F1 score for all our experiments: $\mathrm{F1}_{\mathrm{weighted}} =$

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

	config	prec.	rec.	F1	# dims
CELLPH	**BASE**	.9505	.9497	.9501	NA
	BASE+DOM	.9590	.9590	.9590	100
	BASE+GEN	.9560	.9554	.9557	NA
	BASE+ALL	**.9604**	**.9599**	**.9601**	300
CELLACC	**BASE**	.8571	.8567	.8569	NA
	BASE+DOM	.8723	.8731	.8727	500
	BASE+GEN	.8648	.8649	.8648	NA
	BASE+ALL	**.8806**	**.8812**	**.8809**	300
MSHOES	**BASE**	.8248	.8213	.8230	NA
	BASE+DOM	.8491	.8486	.8488	100
	BASE+GEN	.8376	.8338	.8357	NA
	BASE+ALL	**.8581**	**.8550**	**.8565**	200
WATCHES	**BASE**	.8243	.8210	.8227	NA
	BASE+DOM	.8382	.8384	.8383	200
	BASE+GEN	.8386	.8372	.8379	NA
	BASE+ALL	**.8496**	**.8480**	**.8488**	200
WCLOTH	**BASE**	.8600	.8619	.8609	NA
	BASE+DOM	.8874	.8882	.8878	400
	BASE+GEN	.8752	.8732	.8742	NA
	BASE+ALL	**.8883**	**.8892**	**.8887**	400

Table 6: Table shows the full set of results (weighted precision, recall, and F1) for each of the five e-Commerce categories we experiment with. The last column shows the best (tuned) word2vec dimensionality for the in–domain word vectors.

$\sum_{t\in\{\text{tags}\}} p(t)\mathrm{F1}(t)$, where $p(t)$ is the relative frequency of tag t in the test set and $\mathrm{F1}(t)$ is the F1 score for tag t.

Several trends are clear from the results. First, the combined feature set based on in–domain and out–of–domain word vectors (**BASE+ALL**) gives the best performance for all categories, with a boost of $2+$ percentage points over **BASE** for all categories except CELLPH. Second, most of the improvement over the baseline (**BASE**) is achieved by the in–domain word vector features (**BASE+DOM**). Except for the WATCHES category, the out–of–domain word vector features by themselves are less useful compared to the in–domain vectors. This is not entirely surprising. However, it is worth noting for a couple of reasons: (1) The in–domain data we have, as mentioned earlier, is highly unstructured, and it is not obvious that word vectors trained on such data will be meaningful, let alone useful in a quantitative evaluation like the one we have presented. (2) The in–domain data that we use for word vector training is, in most cases, significantly smaller than the dataset used for training the out–of–domain word vectors. While we directly use the word vectors from Baroni et al. (2014) as our out–of–domain vectors (since they have been shown to perform well across a range of semantic relatedness tasks), in the future it might be worth tuning the out–of–domain word vectors specifically for our task.

In order to gain an understanding of where the distributed word representations are useful, we performed an error analysis on the predictions from our various models. Table 7 shows several different item titles where our trained models differed. The table shows, for example, that the **BASE+DOM** model is able to identify “Movistar” as a `brand` correctly, while the **BASE** model is not. This is interesting because “Movistar” does not appear in our training data at all. However, it does have a representation in our `word2vec` model, and thus the **BASE+DOM** model is able to correctly tag it. The **BASE+DOM** model also correctly tags both tokens in “Red Pocket” as a `brand`, unlike the **BASE+GEN** model, which tags them as `color` and `contract` incorrectly. This shows that the in–domain semantic representation for the token “Red” is more useful compared to its out–of–domain representation. Finally, there are also cases where the out–of–domain semantic representation adds value: “TANGERINE”, for example, is correctly predicted as a `color` by **BASE+ALL**, but not by **BASE+DOM** because it is not present in our in–domain vectors.

6 Related Work

6.1 Word representations

The problem of modeling the meaning of words in text has been approached in various ways including distributional semantics (see Turney and Pantel (2010), Erk (2012) for surveys), word clustering (Brown et al., 1992; Lin and Wu, 2009), and, more recently, distributed representations (Mnih and Hinton, 2007; Collobert and Weston, 2008).

While word clusters and distributional approaches have been shown to be very effective for NER applications (Miller et al., 2004; Lin and Wu, 2009; Ratinov and Roth, 2009; Turian et al., 2010; Dhillon et al., 2011), direct applications of distributed representations to NER systems did not show benefit over Brown clusters (Turian et al., 2010). However, Passos et al. (2014) recently reported performance

	sample titles
BASE+DOM > BASE	New Samsung Galaxy S3 i9300 Unlocked Movistar (brand, no-tag) Claro (brand, no-tag) Vodafone (brand, no-tag) ATT Fido O2 (brand, product) Fido
	VERTU SUPER (no-tag, product) LUXURY CELLPHONE CONSTELLATION AYXTA (product, no-tag) PINK (color, no-tag) WITH RUBBY KEY NEVER USED
BASE+DOM > BASE+GEN	Brand New Nokia 101 100 Unlocked GSM Cellular Phone Phantom (color, no-tag) Black 2 SIM / w MP3
	iPhone 4S STRAIGHT TALK 32GB White Net10 ATT H2 AIO (brand, product) AirVoice (brand, product) Red (brand, color) Pocket (brand, contract) unlocked
BASE+ALL > BASE+DOM	NEW IN BOX SONY ERICSSON W380a W380 BLACK ORANGE TANGERINE (color, no-tag) UNLOCKED GSM Phone
	NEW Unlocked black BlackBerry Bold 9900 gsm cell phone telus (brand, no-tag) rogers (brand, no-tag) koodoo pda

Table 7: A small sample of errors made by our various models on the CELLPH category. The first column shows the models being compared (">" stands for "better than"). The predictions of the models differ for the underlined tokens. In parentheses, the prediction from the correct model is shown first, followed by the prediction of the incorrect model.

comparable to state-of-the-art NER systems using a modified skip-gram model trained to predict membership of words to a domain specific lexicon.

6.2 E-Commerce

E-commerce has recently garnered attention in the natural language processing research community.

Ghani et al. (2006) and Putthividhya and Hu (2011) also address the problem of structuring items in the e-commerce domain through NER and present experimental results on data similar to ours, but do not leverage word vector representations. Mauge et al. (2012) presents an unsupervised approach for identifying attribute names and values from unstructured natural language listings seen in e-commerce sites. Finally, unrelated to NER, Shen et al. (2012) proposed a method for hierarchical classification of product offers which they validated on eBay data.

7 Conclusions

Distributed word representations have been used successfully for improving performance on several natural language processing tasks in the recent past, including the task of named entity recognition (NER). Much of the work, however, has focused on learning these word representations from corpora that consist of relatively well–formed, grammatical language. Moreover, the NER tasks that used these word representations were also based on similar well–formed language. In this work we explore distributed word representations based on e-commerce domain item titles, which are highly unstructured in nature. We also evaluate our constructed word vectors on the task of NER for these item titles.

Our experiments show the following: (1) It is possible to learn useful (as evaluated quantitatively on an NER task) distributed word representations based on unstructured e-commerce item title data. (2) The word representations that we train on a relatively small amount of in–domain data are, in general, more useful than word representations trained on very large out–of–domain data. (3) The combination of in–domain and out–of–domain word representations gives the best result, adding domain–knowledge where necessary, while also using background general knowledge from out–of–domain representations.

Based on our experiments, there are a couple of interesting questions that may be considered for future research. First, we use the most straightforward way of combining in–domain and out–of–domain knowledge – training these word representations separately and using features based on both of them. Whether it is possible to learn better word representations by considering in–domain and out–of–domain data simultaneously at training time is an open question. Second, in our task formulation, the multiple e-commerce categories were trained separately even though they share some semantic tags. This can be improved upon in the future by considering approaches to multi–task learning.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167.

Paramveer Dhillon, Dean P Foster, and Lyle H. Ungar. 2011. Multi-view learning of word embeddings via cca. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 199–207.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explorations*, 8:41–48.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1030–1038.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit.

Karin Mauge, Khash Rohanimanesh, and Jean-David Ruvini. 2012. Structuring e-commerce inventory. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pages 805–814.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 337–342.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 641–648.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL 2014)*, pages 78–86.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *EMNLP*, pages 1557–1567.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155.

Dan Shen, Jean-David Ruvini, and Badrul Sarwar. 2012. Large-scale item categorization for e-commerce. In *CIKM*, pages 595–604.

Jun Suzuki and Hideki Isozaki. 2008. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proceedings of ACL-08: HLT*, pages 665–673.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.