On Deep Learning in Cross-Domain Sentiment Classification[*]

Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani and Roberto Pasolini
Department of Computer Science and Engineering, University of Bologna,
Via Venezia 52, I-47521 Cesena, Italy

Keywords: Transfer Learning, Language Heterogeneity, Sentiment Analysis, Cross-Domain, Big Data.

Abstract: *Cross-domain* sentiment classification consists in distinguishing positive and negative reviews of a target domain by using knowledge extracted and transferred from a heterogeneous source domain. Cross-domain solutions aim at overcoming the costly pre-classification of each new training set by human experts. Despite the potential business relevance of this research thread, the existing ad hoc solutions are still not scalable with real large text sets. Scalable Deep Learning techniques have been effectively applied to *in-domain* text classification, by training and categorising documents belonging to the same domain. This work analyses the cross-domain efficacy of a well-known unsupervised Deep Learning approach for text mining, called Paragraph Vector, comparing its performance with a method based on Markov Chain developed ad hoc for cross-domain sentiment classification. The experiments show that, once enough data is available for training, Paragraph Vector achieves accuracy equivalent to Markov Chain both in-domain and cross-domain, despite no explicit transfer learning capability. The outcome suggests that combining Deep Learning with transfer learning techniques could be a breakthrough of ad hoc cross-domain sentiment solutions in big data scenarios. This opinion is confirmed by a really simple multi-source experiment we tried to improve transfer learning, which increases the accuracy of cross-domain sentiment classification.

1 INTRODUCTION

Understanding people's opinions about products, services, brands and so on is a compelling *sentiment classification* task of valuable importance for operational as well as strategic business decisions. Nevertheless, semantic comprehension of natural language text is definitely arduous, because of its intrinsic ambiguity and context dependence. Both word polarity, namely its positive or negative orientation, and relationships among words have to be taken into account to properly understand the meaning of a sentence. Then the task becomes even more challenging when document-level understanding is required, namely when the overall document polarity has to be discovered. Recently, Deep Learning has given a boost to sentiment classification due to its intrinsic ability in mining hidden relationships in text. Deep Learning approaches are usually more robust and efficient than those based on classical text mining techniques, because their performance typically scales better with dataset size both in terms of accuracy and from the computational point of view.

A classification task consists in using an *in-domain* approach, where documents of the training and test belong to the same domain, for instance classifying a new set of book reviews, after the training on a pre-classified text set, but always of book reviews. However, in practice this is not always feasible because of the missing or insufficient availability of labelled documents to be used for training the model. This is particularly evident in social network posts, such as Facebook, Twitter, LinkedIn etc., and more generally in chats, emails and opinions or reviews in fora, blogs, online press and so on. They are all examples of plain texts, wherein authors can write whatever without strict content constraints. Although there are no labels associated to the plain texts, they have proved to be useful for supporting complex tasks, such as stock market prediction (Domeniconi et al., 2017) and job recommendation (Domeniconi et al., 2016). A solution to the lack of labelled documents is to let a team of human experts pre-classify one or more document sets, so as to have enough data for effectively training algorithms. Unfortunately, the text pre-classification by human experts, though ef-

[*]This work was partially supported by the project "Toreador", funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 688797.

Domeniconi, G., Moro, G., Pagliarani, A. and Pasolini, R.
On Deep Learning in Cross-Domain Sentiment Classification.
DOI: 10.5220/0006488100500060
In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2017) - Volume 1: KDIR*, pages 50-60
ISBN: 978-989-758-271-4

fective to extract reliable knowledge, is a costly activity, in general infeasible for the wide variety and large volume of real big data sources. The *cross-domain learning* thread has been introduced to address these limitations. Basically, after that a knowledge model has been built on a source domain of pre-classified data, for instance from book reviews labelled as positive and negative, the goal of cross-domain learning is to reuse this knowledge in a different unclassified target domain, for instance to distinguish positive and negative unlabelled DVD reviews. The main difference between in-domain and cross-domain is that the latter approach generally requires a *transfer learning* phase, so that the knowledge model built on the source domain could be effectively applied to the target domain. Indeed, language is typically heterogeneous in documents of different domains. For instance, just think that a book can be *engaging* or *boring*, whereas an electrical appliance can be *working* or *damaged*. A human being is able to easily understand that *engaging* and *working* have both positive orientation, while *boring* and *damaged* are negative attributes. On the other hand, this inference is challenging for an automated system: in fact, it could not trivially infer that *working* has a positive meaning if it only knows that *engaging* is positive but the two words never co-occur in the document set used for training. Therefore, transferring the knowledge learnt from the source domain to the target domain is fundamental in cross-domain tasks.

As far as we are concerned, despite the recent success of Deep Learning in several research areas, little effort has been made in cross-domain sentiment classification so far. This work investigates if, and to what extent, Deep Learning algorithms can automatically bridge the inter-domain semantic gap typical of cross-domain sentiment classification, strengthened by their ability to learn syntactic as well as possibly hidden relationships in text. Our research is motivated by the fact that several works (Socher et al., 2013; Le and Mikolov, 2014; Zhang and LeCun, 2015; Tang et al., 2015) pointed out the capability of Deep Learning to learn semantic-bearing word representation, which is typically achieved without supervision, independently of specific domains. To assess the potentiality of Deep Learning in cross-domain sentiment classification, we compare two different approaches: a Markov Chain based method developed by Domeniconi et al. (Domeniconi et al., 2015b), and Paragraph Vector by Le and Mikolov (Le and Mikolov, 2014). The Markov Chain based approach is tailored to cross-domain sentiment classification, where it has achieved the state-of-the-art performance on some benchmark text sets. Paragraph Vector is a well-known unsupervised Deep Learning method that is able to map words into a vector space wherein semantics arises, that is, similar words are nearer than unrelated ones. Although Paragraph Vector has not been designed for cross-domain sentiment classification and does not provide a transfer learning phase, we argue that its ability in learning semantic-bearing word representation could help bridging the inter-domain semantic gap.

To assess this idea, we first perform in-domain experiments, which act as baseline. Then, the cross-domain ability of both approaches is evaluated and discussed. The outcome shows that Paragraph Vector needs very large training sets for learning accurate word and paragraph representations. Anyhow, once enough data is available for training, it achieves accuracy comparable with Markov Chain not only in-domain but also cross-domain, despite not providing any explicit transfer learning mechanism. We strongly believe that if only we were able to combine transfer learning and Deep Learning methods together, there would be a breakthrough of ad hoc cross-domain sentiment solutions in big data scenarios. To validate our opinion, we then propose a really simple multi-source approach, where knowledge is extracted from N heterogeneous source domains and then the resulting model is applied to a target domain. The basic idea is that more variability in instances can be captured when training the model, and this should help the transfer learning capability of Paragraph Vector. In fact, this experiment shows that Paragraph Vector achieves a significant improvement in terms of accuracy in the N sources cross-domain problem. It is important to restate that, although Paragraph Vector does not provide a transfer learning mechanism, it has shown to be able to automatically extract relevant domain-independent information anyway. Moreover, the outcome suggests that the multi-source approach can definitely help transfer learning in cross-domain sentiment classification.

The rest of the paper is organised as follows. After a review of the related literature in Section 2, we outline the main features of the compared methods in Section 3. Then, Section 4 introduces and discusses the performed experiments. Finally, Section 5 points out conclusions and possible future work.

2 STATE OF THE ART

Transfer learning techniques are usually advisable to effectively map knowledge extracted from a *source* domain into a *target* domain. This is particularly useful in *cross-domain* methods, also known as *domain* *adaptation* methods (Daume III and Marcu, 2006), where labelled instances are only available in a source domain but a different target domain is required to be classified. Basically, two knowledge transfer modes have been identified in (Pan and Yang, 2010), namely *instance transfer* and *feature representation transfer*. In order to bridge the inter-domain gap, the former adapts source instances to the target domain, whereas the latter maps source and target features into a different space.

Before the advent of Deep Learning, many approaches have already been attempted to address transfer learning in cross-domain sentiment classification, mostly supervised. Aue and Gamon tried several approaches to adapt a classifier to a target domain: training on a mixture of labelled data from other domains where such data is available, possibly considering just the features observed in the target domain; using multiple classifiers trained on labelled data from different domains; a semi-supervised approach, where few labelled data from the target is included (Aue and Gamon, 2005). Blitzer et al. discovered a measure of domain similarity supporting domain adaptation (Blitzer et al., 2007). Pan et al. advanced a spectral feature alignment to map words from different domains into same clusters, by means of domain-independent terms. These clusters form a latent space that can be used to enhance accuracy on the target domain in a cross-domain sentiment classification problem (Pan et al., 2010). Furthermore, He et al. extended the joint sentiment-topic model by adding prior words sentiment; then, feature and document enrichment were performed by including polarity-bearing topics to align domains (He et al., 2011). Bollegala et al. recommended the adoption of a thesaurus containing labelled data from the source domain and unlabelled data from both the source and the target domains (Bollegala et al., 2013). Zhang et al. proposed an algorithm that transfers the polarity of features from the source domain to the target domain with the independent features as a bridge (Zhang et al., 2015). Their approach focuses not only on the feature divergence issue, namely different features are used to express similar sentiment in different domains, but also on the polarity divergence problem, where the same feature is used to express different sentiment in different domains. Franco et al. used the BabelNet multilingual semantic network to generate features derived from word sense disambiguation and vocabulary expansion that can help both in-domain and cross-domain tasks (Franco-Salvador et al., 2015). Bollegala et al. modelled cross-domain sentiment classification as embedding learning, using objective functions that capture domain-independent features, label constraints in the source documents and some geometric properties derived from both domains without supervision (Bollegala et al., 2016).

On the other hand, the advent of Deep Learning, whose a brilliant review can be found in (LeCun et al., 2015), brought to a dramatic improvement in sentiment classification. Socher et al. introduced the Recursive Neural Tensor Networks to foster single sentence sentiment classification (Socher et al., 2013). Apart from the high accuracy achieved in classification, these networks are able to capture sentiment negations in sentences due to their recursive structure. Dos Santos et al. proposed a Deep Convolutional Neural Network that jointly uses character-level, word-level and sentence-level representations to perform sentiment analysis of short texts (Dos Santos and Gatti, 2014). Kumar et al. presented the Dynamic Memory Network (DMN), a neural network architecture that processes input sequences and questions, forms episodic memories, and generates relevant answers (Kumar et al., 2015). The ability of DMNs in naturally capturing position and temporality allows this architecture achieving the state-of-the-art performance in single sentence sentiment classification over the Stanford Sentiment Treebank proposed in (Socher et al., 2013). Tang et al. introduced Gated Recurrent Neural Networks to learn vector-based document representation, showing that the underlying model outperforms the standard Recurrent Neural Networks in document modeling for sentiment classification (Tang et al., 2015). Zhang and LeCun applied temporal convolutional networks to large-scale data sets, showing that they can perform well without the knowledge of words or any other syntactic or semantic structures (Zhang and LeCun, 2015).

Despite the recent success of Deep Learning in in-domain sentiment classification tasks, few attempts have been made in cross-domain problems. Glorot et al. used the Stacked Denoising Autoencoder introduced in (Vincent et al., 2010) to extract domain-independent features in an unsupervised fashion, which can help transferring the knowledge extracted from a source domain to a target domain (Glorot et al., 2011). However, they relied only on the most frequent 5000 terms of the vocabulary for computational reasons. Although this constraint is often acceptable with small or medium data sets, it could be a strong limitation in big data scenarios, where very large data sets are required to be analysed.

3 METHODS

This Section outlines the main features of the two methods that are compared in this work, namely Paragraph Vector (referred as PV hereinafter), proposed in (Le and Mikolov, 2014), and a Markov Chain (referred as MC hereinafter) based algorithm introduced in (Domeniconi et al., 2015b) and extended in (Domeniconi et al., 2015a).

The former is an unsupervised Deep Learning technique that aims to solve the weaknesses of the bag-of-words model. Alike bag-of-words, PV learns fixed-length feature representation from variable-length pieces of texts, such as sentences, paragraphs, and documents. However, bag-of-words features lose the ordering of the words and also do not capture their semantics. For example, ”good”, ”robust” and ”town” are equally distant in the feature space, despite ”good” should be closer to ”robust” than ”town” from the semantic point of view. The same holds for the bag-of-n-grams model, because it suffers from data sparsity and high dimensionality, although it considers the word order in short context. On the other hand, PV intrinsically handles the word order by representing each document by a dense vector, which is trained to predict words in the document itself. More precisely, the paragraph vector is concatenated with some word vectors from the same document to predict the following word in the given context. The paragraph token can be thought of as another word that acts as a memory that remembers what is missing from the current context. For this reason, this model, represented in Figure 1, is called the Distributed Memory Model of Paragraph Vector (PV-DM).

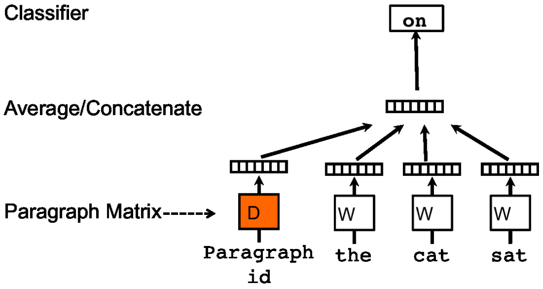

Figure 1: The figure (Le and Mikolov, 2014) shows a framework for learning the Distributed Memory Model of Paragraph Vector (PV-DM). With respect to word vectors, an additional paragraph token is mapped to a vector via matrix D. In this model, the concatenation or average of this vector with a context of three words is used to predict the fourth word. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

Another way to learn the paragraph vector is to ignore the context words in the input, but force the model to predict words randomly sampled from the paragraph in the output. Actually this means that at each iteration of stochastic gradient descent, a text window is sampled, then a random word is sampled from the text window and a classification task is formed given the Paragraph Vector. This version of the Paragraph Vector, shown in Figure 2, is called the Distributed Bag of Words version (PV-DBOW).

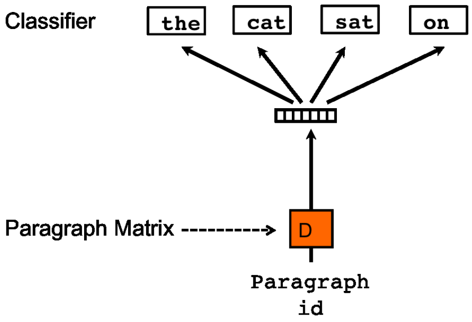

Figure 2: The figure (Le and Mikolov, 2014) shows the Distributed Bag of Words version of Paragraph Vector (PV-DBOW). The paragraph vector is trained to predict the words in a small window.

Both word vectors and paragraph vectors are trained by means of the stochastic gradient descent and backpropagation (Williams and Hinton, 1986).

Sentiment classification requires sequential data to be handled, because the document semantics is typically affected by the word order. PV is shown to be able to learn vector representation for such sequential data, becoming a candidate technique for sentiment classification. We have already stated the PV learns fixed-length feature representation from variable-length pieces of texts, dealing with any kind of plain text, from sentences to paragraphs, to whole documents. Though, this aspect is just as relevant as exactly knowing how many of these features are actually required to learn accurate models. The feature vectors have dimensions in the order of hundreds, much less than bag-of-words based representations, where there is one dimension for each word in a dictionary. The consequence is that either the bag-of-words models cannot be used for representing very large data sets due to the huge number of features or a feature selection is needed to reduce dimensionality. Feature selection entails information loss, beyond requiring parameter tuning to choose the right number of features to be selected. The fact that PV is not affected by the curse of dimensionality suggests that the underlying method is not only scalable just like an algorithm should be when dealing with large data sets, but it also entirely preserves information by increasing the data set size.

(Le and Mikolov, 2014) showed that Paragraph

Vector achieves brilliant in-domain sentiment classification results, but no cross-domain experiment has been conducted. Nevertheless, some characteristics of PV make it appropriate for cross-domain sentiment classification, where the language is usually heterogeneous across domains. PV is very powerful in modelling syntactic as well as hidden relationships in plain text without any kind of supervision. Moreover, words are mapped to positions in a vector space wherein the distance between vectors is closely related to their semantic similarity. The capability of extracting both word semantics and word relationships in an unsupervised fashion makes it appealing to test whether PV is able to automatically manage transfer learning. For this purpose, a comparison with a Markov Chain based method tailored to this task (Domeniconi et al., 2015b) will be shown in Section 4.

As described in (Le and Mikolov, 2014), in order to use the available labelled data, each subphrase is treated as an independent sentence and the representations for all the subphrases in the training set are learnt. After learning the vector representations for training sentences and their subphrases, they are fed to a logistic regression to learn a predictor of the sentiment orientation. At test time, the vector representation for each word is frozen, and the representations for the sentences are learnt using the stochastic gradient descent. Once the vector representations for the test sentences are learnt, they are fed through the logistic regression to predict the final label.

Alike PV, MC can handle sentences, paragraphs and documents, but it is much more affected by the curse of dimensionality, because it is based on a dense bag-of-words model. Feature selection is often advisable to mitigate this issue, or even necessary with very large data sets, typically containing million or billion words. Basically, only the k most significant terms according to a given scoring function are kept. The basic idea of the MC based approach consists in modelling term co-occurrences: the more terms co-occur in documents the more their connection are stronger. The same strategy could be followed to model the polarity of a given term: the more terms are contained in positive (negative) documents the more they tend to be positive (negative). Following this idea, terms and classes are represented as states of a Markov Chain, whereas term-term and term-class relationships are modelled as transitions between these states. Thanks to this representation, MC is able to perform both sentiment classification and transfer learning. It is pretty easy to see that MC can be used as a classifier, because classes are reachable from terms at each state transition in the Markov Chain, since each edge models a term-class relationship. Instead, it is less straightforward to understand why it is also able to perform transfer learning. The assumption the method relies on is that there exists a subset of common terms between the source and target domains that act as a bridge between domain specific terms, allowing and supporting transfer learning. Dealing with this assumption, at each state transition in the Markov Chain, sentiment information can flow from the source-specific to the target-specific terms passing through the layer of shared terms (Figure 3). The information flow is possible by exploiting the edges in the Markov Chain that, as previously stated, represent term-term relationships.

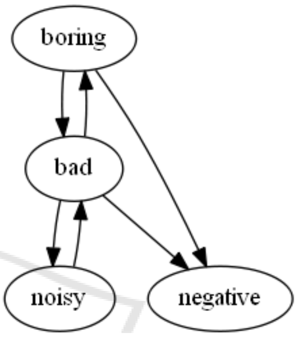

Figure 3: The figure (Domeniconi et al., 2015b) shows transfer learning in the Markov Chain from a book specific term like *boring* to an electrical appliance specific term like *noisy* through a common term like *bad*. ...

Actually, the classification process usually works in the opposite direction, i.e. from the target-specific to the source-specific terms, and goes on while the class states are eventually reached. For instance, say that a review from the target domain only contains target-specific terms. None of these terms is connected to the classes, but they are connected to some terms within the shared terms, which in turn are connected to some source-specific terms. Finally, both the shared and source-specific terms are connected to the classes. Therefore, starting from some target-specific terms, the Markov Chain performs firstly transfer learning and then sentiment classification. It is important to remark that the transfer learning mechanism is not an additional step to be added in cross-domain tasks; on the contrary, it is intrinsic to the Markov Chain algorithm.

Careful readers can find further details on the described approaches in (Le and Mikolov, 2014; Domeniconi et al., 2015b; Domeniconi et al., 2015a).

4 EXPERIMENTS

This Section presents some experiments to show whether outstanding unsupervised techniques as Paragraph Vector are suitable for cross-domain sentiment classification, despite no explicit mechanism to manage transfer learning. The underlying investigation also gives users insights into the awkward choice of the most suitable algorithm for a given problem, with reference to the amount of data available for training.

The Markov Chain based method has been implemented in a custom Java-based framework. Instead, for Paragraph Vector we relied on *gensim* (Rehurek and Sojka, 2010), a Python-based open sourced and freely available framework[2]. In particular, all tests have been performed by using its 0.12.4 software release. Apart from the two main approaches compared, we also employed Naïve Bayes (NB) as baseline for the experiments. The Naïve Bayes implementation is from the 3.9.1 software release of the Weka (Frank et al., 2005) workbench.

4.1 Setup

In order for the results to be comparable, we used a common benchmark data set, namely, a collection of Amazon reviews[3] about Books (B), Movies (M), Electronics (E) and Clothing-Shoes-Jewelry (J). Each domain contains plain English reviews along with their labels, namely a score from 1 (very negative) to 5 (very positive). We mapped the reviews whose scores were 1 and 2 to the negative category, those whose scores were 4 and 5 to the positive one, discarding those whose score was 3 that were likely to express a neutral sentiment orientation.

For the sake of assessing the effectiveness of the algorithms by varying the amount of labelled data available for training, we tested source-target partitions with three different orders of magnitude, always preserving the source-target ratio, i.e. 80%-20%, and the balancing between positive and negative instances. The smallest data set counts 1600 instances as the training set and 400 as the test set; the medium 16000 and 4000 respectively; and the largest 80000 and 20000 respectively. For each examined source-target combination, accuracy has been chosen as performance measure, namely, the percentage of correctly classified instances. This is a typical choice in a balanced binary classification problem, where there is an even number of instances for the 2 categories considered. Moreover, results have been averaged on 10 different training-test partitions to reduce the variance, that is, the sensitivity to small fluctuations in the training set.

For all the following experiments we used the Distributed Bag of Words version of Paragraph Vector (PV-DBOW) presented in (Le and Mikolov, 2014), choosing 100-dimensional feature vectors, considering 10 words in the window size, ignoring words occurring in just one document and applying negative sampling with 5 negative samples. Moreover, we set the initial learning rate to 0.025, letting it linearly drop to 0.001 in 30 epochs. For further details on the parameters, careful readers could refer to (Le and Mikolov, 2014; Mikolov et al., 2013). To accomplish sentiment classification, the positive or negative orientation of the reviews has been predicted by means of a logistic classifier, whose regression coefficients have been estimated employing the Newton-Raphson method.

Concerning the MC based method, we relied on the technique described in (Domeniconi et al., 2015b). Firstly, the relative frequency of terms in documents has been chosen as term weighting. Then, feature selection was required to mitigate the curse of dimensionality because MC is based on a dense bag-of-words model, as explained in Section 3. The features have been selected by means of χ^2 scoring function. After a tiny tuning, we chose 750, 10000 and 25000 terms for the small, medium and large data sets respectively. The Markov Chain has been built including the selected terms only. As already explained in Section 3, the more terms co-occur in reviews the more their connection are stronger. Likewise, the more terms are contained in positive (negative) reviews the more they tend to be positive (negative). For further details, careful readers could refer to (Domeniconi et al., 2015b; Domeniconi et al., 2015a).

Finally, the Naïve Bayes algorithm has been run with default parameters after the same feature selection performed for MC, namely, 750, 10000 and 25000 terms for the small, medium and large data sets respectively by means of χ^2 scoring function.

Three experiments will be shown below. The first focuses on in-domain sentiment classification, namely, where the algorithms are tested on a set of reviews from the same domain used for training. In this way, it is possible to evaluate how the performance varies with respect to the amount of training data. The second experiment focuses on cross-domain sentiment classification, where transfer learning is usually needed to handle the heterogeneity of language across domains. The test assesses the capability of Paragraph Vector in automatically bridging the inter-domain semantic gap, without providing any explicit transfer learning mechanism. The last experi-

[2] http://nlp.fi.muni.cz/projekty/gensim/

[3] http://jmcauley.ucsd.edu/data/amazon/

ment shows a simple multi-source approach, in order to analyse whether this positively affects the performance of Paragraph Vector in cross-domain sentiment classification.

4.2 In-Domain Results

The first experiment assesses the in-domain performance of the algorithms. Table 1 shows the results over the 4 domains of the Amazon dataset, namely Books (B), Movies (M), Electronics (E) and Clothing-Shoes-Jewelry (J).

The first outcome that catches the eye is that PV requires much more training data than MC in order to perform well, as it is even clearer by observing Figure 4. Indeed, although it achieved brilliant results as stated by Le and Mikolov (Le and Mikolov, 2014), it underperforms MC and performs slightly better than Naïve Bayes on average when small data sets are involved. This is not completely surprising because Deep Learning techniques typically require very large training sets to learn models that are able to generalise over new test instances. On the other hand, PV scales very well in terms of accuracy when the model is learnt from very large labelled text sets. In fact, accuracy boosts from 75.44% in the small text set to 77.63% in the medium one, and it is even 84.93% on average in the largest. Careful readers could have noted that, when enough training data is available, the accuracy of PV has low deviation from the mean in each of the four domains. This proves that PV is a robust approach, which is effective independetly of the particular domain analysed.

Apart from what has been just stated about PV, it is noteworthy to point out that the accuracy achieved by MC is pretty stable by increasing the amount of training data. This outcome suggests that both MC and PV could be candidate methods in the analysis of very large data sets. However, a tiny feature selection phase is always demanded by MC before learning the model to reduce dimensionality and let the method be applicable to new data sets.

4.3 Cross-Domain Results

The second experiment is about cross-domain sentiment classification. The goal is to assess whether Paragraph Vector is able to bridge the semantic gap between the source and target domains, despite no explicit transfer learning mechanism. For this purpose, we compare them with the Markov Chain based method in all source-target configurations of the Amazon datasets, namely $B \to E$, $B \to M$, $B \to J$, $E \to B$, $E \to M$, $E \to J$, $M \to B$, $M \to E$, $M \to J$, $J \to B$, $J \to E$, $J \to M$. As in the previous experiment, Naïve Bayes is used as baseline. The results of the comparison are shown in Table 2 and in Figure 5.

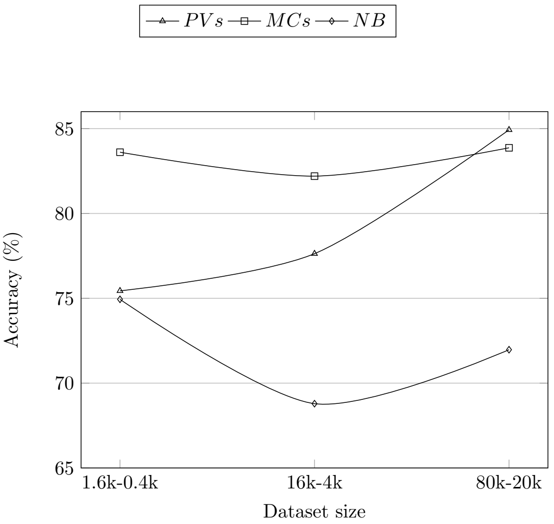

Figure 4: Accuracy achieved by the compared methods on average in the in-domain sentiment classification task, by varying the dataset size. Nk-Mk means that the experiment has been performed by using N*1000 instances as the training set and M*1000 instances as the test set.

As expected, the accuracy of the baseline algorithm is very low on average, because Naïve Bayes has no transfer learning capability. MC performs much better than Naïve Bayes on average, thanks to the transfer learning mechanism described in Section 3. Differently from the in-domain problem, where the accuracy achieved by MC is stable with respect to the dataset size, here its accuracy improves by increasing the number of training examples, as shown in Figure 5. This means that MC requires a large amount of training data in order to effectively transfer the knowledge extracted from the source domain to the target domain.

Though, the most surprising outcome is certainly the accuracy obtained by PV, which is comparable with MC on average and even better in the analysis of the smallest data sets. This is actually astonishing if we think that PV does not provide for a transfer learning phase. It could be explained by considering that PV is instrinsically able to handle the word order, because each document is represented by a dense vector, which is trained to predict the following word in the document itself. Training the model, terms are mapped into a vector space where the distance between them is related to their semantics. For example, the distance between ”good” and ”robust” is less than the distance between either of these terms and ”town”. The fact that PV learns vector representa-

Table 1: Comparison between PV and MC in sentiment classification, using NB as baseline. Nk-Mk means that the experiment has been performed by using N*1000 instances as the training set and M*1000 instances as the test set. $X \rightarrow Y$ means that the model has been learnt on reviews from domain X and then applied to different reviews from domain Y.

	1.6k-0.4k			16k-4k			80k-20k		
Domain(s)	*PV*	*MC*	*NB*	*PV*	*MC*	*NB*	*PV*	*MC*	*NB*
In-domain experiments									
$B \rightarrow B$	67.25%	79.25%	78.25%	75.40%	81.90%	63.30%	84.74%	83.84%	66.36%
$M \rightarrow M$	79.75%	91.23%	74.50%	74.87%	82.43%	70.10%	84.11%	80.23%	71.30%
$E \rightarrow E$	79.25%	92.00%	79.50%	80.15%	80.72%	69.53%	85.61%	84.41%	76.41%
$J \rightarrow J$	75.50%	71.97%	67.50%	80.08%	83.76%	72.23%	85.25%	86.98%	73.81%
Average	**75.44%**	**83.61%**	**74.94%**	**77.63%**	**82.20%**	**68.79%**	**84.93%**	**83.87%**	**71.97%**

Table 2: Comparison between PV and MC in cross-domain sentiment classification, using NB as baseline. Nk-Mk means that the experiment has been performed by using N*1000 instances as the training set and M*1000 instances as the test set. $X \rightarrow Y$ means that the model has been learnt on reviews from the source domain X and then applied to reviews from the target domain Y.

	1.6k-0.4k			16k-4k			80k-20k		
Domain(s)	*PV*	*MC*	*NB*	*PV*	*MC*	*NB*	*PV*	*MC*	*NB*
Cross-domain experiments (*source* → *target*)									
$B \rightarrow E$	70.75%	69.29%	67.50%	67.27%	71.22%	55.85%	73.24%	74.05%	58.77%
$B \rightarrow M$	66.75%	70.85%	70.50%	80.25%	79.32%	61.15%	81.97%	79.01%	61.99%
$B \rightarrow J$	73.25%	79.70%	63.75%	70.60%	71.83%	53.38%	74.87%	75.99%	54.92%
$E \rightarrow B$	74.00%	54.00%	64.50%	78.80%	80.10%	65.43%	76.87%	79.19%	66.15%
$E \rightarrow M$	71.50%	56.75%	70.75%	76.17%	76.20%	64.43%	76.86%	77.15%	66.06%
$E \rightarrow J$	82.75%	74.25%	72.00%	79.47%	80.49%	63.23%	80.80%	81.91%	73.09%
$M \rightarrow B$	74.75%	65.75%	65.25%	85.55%	86.05%	76.55%	85.21%	83.81%	69.05%
$M \rightarrow E$	71.75%	68.18%	65.25%	75.32%	77.10%	66.35%	74.79%	72.87%	63.94%
$M \rightarrow J$	82.25%	81.95%	63.25%	73.45%	74.86%	62.03%	76.96%	78.58%	67.26%
$J \rightarrow B$	66.25%	75.25%	62.50%	69.62%	80.55%	64.48%	76.53%	78.55%	65.88%
$J \rightarrow E$	76.50%	80.60%	75.75%	78.55%	79.76%	68.90%	80.08%	81.79%	70.88%
$J \rightarrow M$	74.25%	81.25%	72.50%	70.77%	74.30%	63.25%	76.07%	77.93%	66.27%
Average	**73.73%**	**71.49%**	**67.79%**	**75.49%**	**77.65%**	**63.75%**	**77.85%**	**78.40%**	**65.36%**

tions without any kind of supervision is probably what could have helped more bridging the inter-domain semantic gap. Some readers could object that the average cross-domain accuracy in the smallest data sets is surprising. In fact, while we stated in 4.2 that PV requires big training sets to perform well, PV outperforms MC on average in the respective cross-domain configuration, where the two methods achieve 73.73% and 71.49% accuracy respectively. The explanation of this outcome has to be found in the concept of supervision. MC includes a transfer learning mechanism, which relies on labelled data to transfer semantics from the source domain to the target domain. Therefore, when few labelled data is available, the algorithm does not achieve high accuracy on target domains. On the other hand, PV does not handle transfer learning explicitly and relies on an unsupervised approach to map terms in a vector space, i.e. the feature space. For this reason, PV is less affected than MC by the change of domain.

This experiment has shown that PV is so able to generalise that it could even foster a challenging task as cross-domain sentiment classification, especially if pre-trained without supervision to learn fixed-length vector representations of terms. We argue that unsupervised pre-training of Deep Learning algorithms, if opportunely combined with a proper transfer learning approach, can be a breakthrough of ad hoc cross-domain sentiment solutions in big data scenarios.

4.4 Multi-Source Results

The third experiment evaluates the impact of a multi-source approach on the transfer learning capability of Paragraph Vector. Multi-source basically means that 3 out of 4 domains are used to train the model, which is then tested on the remaining domain. For instance, the model is built on Books, Electronics and Movies, and then applied to Jewelry. Such a configuration is referred as $* \rightarrow J$, and the others are assembled in the same way. This still is a cross-domain sentiment classification problem, because the model is learnt on labelled data from some domains but its performance is evaluated on a different unlabelled domain. The only difference between the single-source Paragraph Vector ($1S-PV$) and the multi-source Paragraph Vector

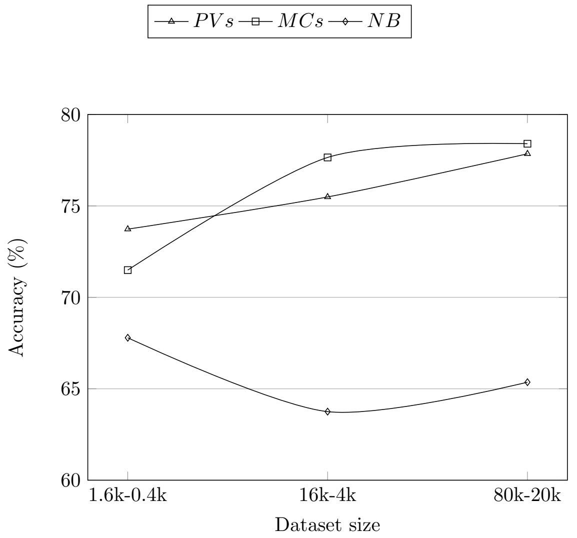

Figure 5: Accuracy achieved by the compared methods on average in the cross-domain sentiment classification task, by varying the dataset size. Nk-Mk means that the experiment has been performed by using N*1000 source instances as the training set and M*1000 target instances as the test set.

($MS-PV$) is that the latter relies on heterogeneous data sources when training the model.

The experiment has been performed to support our claim that Deep Learning and transfer learning solutions can, if combined, break through cross-domain sentiment classification. The rationale is that, training the model on heterogeneous domains, more variability in instances can be captured and Paragraph Vector could automatically learn how to handle the language heterogeneity, improving its transfer learning capability and, as a consequence, its cross-domain performance.

Table 3 and Figure 6 report the comparison between the single-source and multi-source Paragraph Vector in cross-domain sentiment classification. The training on multiple heterogeneous domains allows Paragraph Vector learning better semantic-bearing word representation than using a single domain only. It is pretty easy to see that $MS-PV$ outperform $1S-PV$ on average, achieving accuracy from 2% to 3% higher independently of the dataset size. This proves that even a simple gimmick as the multi-source approach is effective to increase the accuracy of PV in cross-domain tasks, despite it has not been designed to explicitly handle transfer learning. Furthermore, the outcome supports our claim that combining Deep Learning techniques as Paragraph Vector with more advanced transfer learning solutions could break through cross-domain sentiment classification.

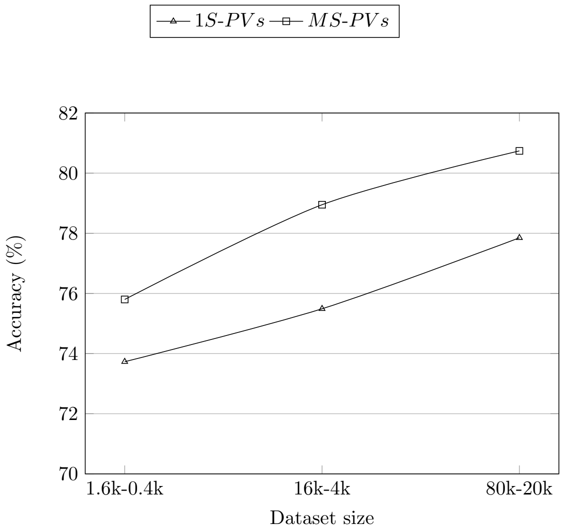

Figure 6: Accuracy achieved by the compared methods on average in the multi-source cross-domain sentiment classification task, by varying the dataset size. Nk-Mk means that the experiment has been performed by using N*1000 source instances as the training set and M*1000 target instances as the test set. Actually, the number of training instances is such that there is an even number of examples from each of the source domains.

5 CONCLUSIONS AND FUTURE WORK

The cross-domain sentiment classification distinguishes positive and negative reviews of a domain, such as car reviews, by exploiting and transferring the knowledge extracted from another domain, generally heterogeneous in language, such as pre-classified electronics reviews.

The goal of this work was to experimentally evaluate if, and to what extent, a well-known Deep Learning algorithm, not designed for cross-domain classification, can compete with ad hoc solutions based on transfer learning techniques. We compared an unsupervised Deep Learning technique known as Paragraph Vector (PV), unprovided for explicit transfer learning capability, with a state-of-the-art Markov Chain based algorithm (MC), tailored to cross-domain sentiment classification. The major outcome is that the Deep Learning algorithm is able to extract generalised knowledge in an unsupervised fashion, so as to bridge the inter-domain semantic gap and achieve comparable performance with MC. This result persuades us that, if transfer learning solutions were explicitly added to unsupervised pre-trained Deep Learning approaches as Paragraph Vector, there would be a breakthrough of ad hoc cross-

Table 3: Comparison between single-source PV (referred as $1S-PV$) and multi-source PV (referred as $MS-PV$) in cross-domain sentiment classification. $*\to Y$ means that the model has been applied to reviews from the target domain Y, after learning on the others except from Y. Nk-Mk means that the experiment has been performed by using N*1000 source instances as the training set and M*1000 target instances as the test set. Actually, the number of training instances is such that there is an even number of examples from each of the source domains.

	1.6k-0.4k		16k-4k		80k-20k	
Domain(s)	$1S-PV$	$MS-PV$	$1S-PV$	$MS-PV$	$1S-PV$	$MS-PV$
Multi-source experiments						
$*\to B$	71.67%	76.85%	77.99%	78.22%	79.54%	81.38%
$*\to E$	73.00%	75.08%	73.71%	78.04%	76.04%	78.46%
$*\to J$	79.42%	75.15%	74.51%	78.79%	77.54%	81.05%
$*\to M$	70.83%	76.10%	75.73%	80.73%	78.30%	82.06%
Average	**73.73%**	**75.80%**	**75.49%**	**78.95%**	**77.85%**	**80.74%**

domain methods. Furthermore, since Paragraph Vector can learn fixed-length feature representation from variable-length pieces of texts and, hence, it is not threaten by the curse of dimensionality, the breakthrough will also involve big data scenarios.

To certify our viewpoint, we proposed a really simple multi-source approach, where knowledge is extracted from N heterogeneous source domains and the resulting model is applied to a different target domain. The idea was that the model could capture more variability in instances if trained on more than a single source domain. Our hypothesis has been confirmed by the experiments, which have shown that accuracy increases of $2-3\%$ on average when training Paragraph Vector on multiple source domains rather than on a single source domain. The boost in terms of accuracy is independent of the dataset size. This supports our belief that the breakthrough, which is feasible by combining Deep Learning and transfer learning, will also involve big data scenarios, where very large data sets are usually required to be analysed.

Future work will focus on combining transfer learning approaches with Deep Learning solutions as Paragraph Vector and other techniques. A possible option is to use a semi-supervised approach: basically, after the training on one or more source domains, a fine-tuning phase is performed, where the model is refined on few instances of the target domain before applying it to classify new target examples. Another viable alternative is to combine Paragraph Vector with the Markov Chain based method, exploiting the advantages of both approaches. On the one hand, PV is able to learn word semantics without supervision; on the other hand, MC provides a transfer learning mechanism to bridge the gap between the source and target domains in cross-domain sentiment classification.

ACKNOWLEDGEMENTS

Heartfelt thanks to Professor Claudio Sartori for the discussions and exchange of ideas about Data Mining, Machine Learning and Deep Learning, which pave the way for this work.

REFERENCES

Aue, A. and Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*.

Blitzer, J., Dredze, M., Pereira, F., et al. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 7, pages 440–447.

Bollegala, D., Mu, T., and Goulermas, J. Y. (2016). Cross-domain sentiment classification using sentiment sensitive embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):398–410.

Bollegala, D., Weir, D., and Carroll, J. (2013). Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1719–1731.

Daume III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

Domeniconi, G., Moro, G., Pagliarani, A., Pasini, K., and Pasolini, R. (2016). Job recommendation from semantic similarity of linkedin users' skills. In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 270–277.

Domeniconi, G., Moro, G., Pagliarani, A., and Pasolini, R. (2015a). Cross-domain sentiment classification via polarity-driven state transitions in a markov model. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 118–138. Springer.

Domeniconi, G., Moro, G., Pagliarani, A., and Pasolini, R. (2015b). Markov chain based method for in-domain and cross-domain sentiment classification. In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 127–137. Scitepress.

Domeniconi, G., Moro, G., Pagliarani, A., and Pasolini, R. (2017). Learning to predict the stock market dow jones index detecting and mining relevant tweets. In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*.

Dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78.

Franco-Salvador, M., Cruz, F. L., Troyano, J. A., and Rosso, P. (2015). Cross-domain polarity classification using a knowledge-enhanced meta-classifier. *Knowledge-Based Systems*, 86:46–56.

Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., and Trigg, L. (2005). Weka. *Data Mining and Knowledge Discovery Handbook*, pages 1305–1314.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

He, Y., Lin, C., and Alani, H. (2011). Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 123–131. Association for Computational Linguistics.

Kumar, A., Irsoy, O., Su, J., Bradbury, J., English, R., Pierce, B., Ondruska, P., Gulrajani, I., and Socher, R. (2015). Ask me anything: Dynamic memory networks for natural language processing. *CoRR, abs/1506.07285*.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web - WWW 2010*, pages 751–760. Association for Computing Machinery (ACM).

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Rehurek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. University of Malta.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 1631–1642.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432. Association for Computational Linguistics (ACL).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

Williams, D. and Hinton, G. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–538.

Zhang, X. and LeCun, Y. (2015). Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.

Zhang, Y., Hu, X., Li, P., Li, L., and Wu, X. (2015). Cross-domain sentiment classification-feature divergence, polarity divergence or both? *Pattern Recognition Letters*, 65:44–50.