Transfer learning in sentiment classification with deep neural networks

Metadata of the chapter that will be visualized in SpringerLink

Book Title	Knowledge Discovery, Knowledge Engineering and Knowledge Management	
Series Title		
Chapter Title	Transfer Learning in Sentiment Classification with Deep Neural Networks	
Copyright Year	2019	
Copyright HolderName	Springer Nature Switzerland AG	
Author	Family Name	**Pagliarani**
	Particle	
	Given Name	**Andrea**
	Prefix	
	Suffix	
	Role	
	Division	Department of Computer Science and Engineering
	Organization	University of Bologna
	Address	Via Cesare Pavese, 47522, Cesena, Italy
	Email	andrea.pagliarani12@unibo.it
Corresponding Author	Family Name	**Moro**
	Particle	
	Given Name	**Gianluca**
	Prefix	
	Suffix	
	Role	
	Division	Department of Computer Science and Engineering
	Organization	University of Bologna
	Address	Via Cesare Pavese, 47522, Cesena, Italy
	Email	gianluca.moro@unibo.it
Author	Family Name	**Pasolini**
	Particle	
	Given Name	**Roberto**
	Prefix	
	Suffix	
	Role	
	Division	Department of Computer Science and Engineering
	Organization	University of Bologna
	Address	Via Cesare Pavese, 47522, Cesena, Italy
	Email	roberto.pasolini@unibo.it
Author	Family Name	**Domeniconi**
	Particle	
	Given Name	**Giacomo**
	Prefix	
	Suffix	

Role	
Division	
Organization	IBM - Watson Research Center
Address	1101 Kitchawan Road, Yorktown Heights, NY, 10598, USA
Email	giacomo.domeniconi1@ibm.com

Abstract

Cross-domain sentiment classifiers aim to predict the polarity (i.e. sentiment orientation) of target text documents, by reusing a knowledge model learnt from a different source domain. Distinct domains are typically heterogeneous in language, so that transfer learning techniques are advisable to support knowledge transfer from source to target. Deep neural networks have recently reached the state-of-the-art in many NLP tasks, including in-domain sentiment classification, but few of them involve transfer learning and cross-domain sentiment solutions. This paper moves forward the investigation started in a previous work [1], where an unsupervised deep approach for text mining, called Paragraph Vector (PV), achieved cross-domain accuracy equivalent to a method based on Markov Chain (MC), developed ad hoc for cross-domain sentiment classification. In this work, Gated Recurrent Unit (GRU) is included into the previous investigation, showing that memory units are beneficial for cross-domain when enough training data are available. Moreover, the knowledge models learnt from the source domain are tuned on small samples of target instances to foster transfer learning. PV is almost unaffected by fine-tuning, because it is already able to capture word semantics without supervision. On the other hand, fine-tuning boosts the cross-domain performance of GRU. The smaller is the training set used, the greater is the improvement of accuracy.

Keywords (separated by '-')

Transfer learning - Cross-domain - Deep learning - Fine-tuning - Sentiment analysis - Big Data

Transfer Learning in Sentiment Classification with Deep Neural Networks

Andrea Pagliarani[1], Gianluca Moro[1(✉)], Roberto Pasolini[1], and Giacomo Domeniconi[2]

[1] Department of Computer Science and Engineering, University of Bologna, Via Cesare Pavese, 47522 Cesena, Italy
{andrea.pagliarani12,gianluca.moro,roberto.pasolini}@unibo.it
[2] IBM - Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, NY 10598, USA
giacomo.domeniconi1@ibm.com

Abstract. Cross-domain sentiment classifiers aim to predict the polarity (i.e. sentiment orientation) of target text documents, by reusing a knowledge model learnt from a different source domain. Distinct domains are typically heterogeneous in language, so that transfer learning techniques are advisable to support knowledge transfer from source to target. Deep neural networks have recently reached the state-of-the-art in many NLP tasks, including in-domain sentiment classification, but few of them involve transfer learning and cross-domain sentiment solutions. This paper moves forward the investigation started in a previous work [1], where an unsupervised deep approach for text mining, called Paragraph Vector (PV), achieved cross-domain accuracy equivalent to a method based on Markov Chain (MC), developed ad hoc for cross-domain sentiment classification. In this work, Gated Recurrent Unit (GRU) is included into the previous investigation, showing that memory units are beneficial for cross-domain when enough training data are available. Moreover, the knowledge models learnt from the source domain are tuned on small samples of target instances to foster transfer learning. PV is almost unaffected by fine-tuning, because it is already able to capture word semantics without supervision. On the other hand, fine-tuning boosts the cross-domain performance of GRU. The smaller is the training set used, the greater is the improvement of accuracy.

Keywords: Transfer learning · Cross-domain · Deep learning · Fine-tuning · Sentiment analysis · Big Data

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. We thank NVIDIA Corporation for the donated Titan GPU used in this work.

A. Fred et al. (Eds.): IC3K 2017, CCIS 976, pp. 1–23, 2019.
https://doi.org/10.1007/978-3-030-15640-4_1

1 Introduction

Sentiment analysis deals with the computational treatment of opinion, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes (a survey is in [2]). The task is technically challenging but very useful in practice. For instance, companies always want to know customer opinions about their products and services.

When an understanding of plain text document polarity (e.g. positive, negative or neutral orientation) is required, sentiment classification is involved. This supervised approach aims to learn a model from a labelled training set of documents, then to apply it to an unlabelled test set, whose sentiment orientation has to be found. The typical approach to sentiment classification assumes that both the training set and the test set deal with the same topic. For example, a model is learnt on a set of board game reviews and applied to a distinct set of reviews, but always about board games. This modus operandi, known as in-domain sentiment classification, guarantees optimal performance provided that documents from the same domain are semantically similar. Unluckily, this approach is often inapplicable in practice, given that most documents are normally unlabelled. Tweets, blogs, fora, chats, emails, public repositories, social networks could bear opinions, and have been proved to support complex tasks, such as stock market prediction [3], job recommendation [4] and genomics [5]. However, no information is available on whether such opinions are positive, negative or neutral. Text categorisation by human experts is the only way to deal with such a problem in order to learn an in-domain sentiment classifier. This method becomes infeasible as soon as very large text sets are required to be labelled, like for instance in big data scenarios.

Transfer learning addresses exactly these limitations, paving the way for model reuse [6]. While these methods are used in image matching [7], genomic prediction [8–10] and many other contexts, their most common application is perhaps in text document categorisation. Basically, a knowledge model, once learnt on a source domain, can be applied to classify document polarity in a distinct target domain. For instance, a model built on a set of labelled documents about board games (i.e. source domain) could be employed for the categorisation of a set of unlabelled documents about electrical appliances (i.e. target domain). The practical implications of model reuse made cross-domain learning a hot research thread. The biggest obstacle to learning an effective cross-domain sentiment classifier is the language heterogeneity in documents of different domains. Just think that a board game can be *engaging* or *dull*, whereas an electrical appliance can be *working* or *broken*. In such cases, transfer learning (or knowledge transfer) techniques may help solving the problem, so that the knowledge extracted from the source is available to classify the target.

To the best of our knowledge, transfer learning has rarely been applied to sentiment classification with deep learning techniques, despite their success in other research areas. Several works [11–14] motivate such an investigation, pointing out the ability of deep approaches to learn semantic-bearing word representation, typically without supervision, independently of domains.

Our previous work [1] has begun the study by comparing a well-known unsupervised deep learning technique, namely Paragraph Vector [12], with a Markov Chain approach [15,16], tailored to cross-domain sentiment classification. When enough data are available for training, Paragraph Vector achieves accuracy comparable with Markov Chain, despite no explicit transfer learning mechanism. The outcome suggested that cross-domain solutions could be dramatically improved by combining deep learning with transfer learning techniques. The multi-source approach, proposed to validate this intuition, boosted the cross-domain accuracy from 2% to 3% depending on the configuration.

This paper carries on with the investigation on deep learning in cross-domain sentiment classification, by including Gated Recurrent Unit (GRU) [17] in the comparison. GRU is a deep architecture, evolution of LSTM, able to adaptively capture dependencies of different time scales. Similarly to Paragraph Vector, GRU does not provide any explicit transfer learning mechanism. Apart from supporting the outcome of our previous work with the inclusion of another outstanding deep learning technique, this paper also shows the impact of fine-tuning on cross-domain sentiment classification. Fine-tuning is an explicit transfer learning mechanism, where a small sample of target instances is used to tune the parameters of a model learnt from the source domain.

Experiments have been carried out to compare GRU with both PV and MC in cross-domain sentiment classification. The same benchmark text sets have been used to assess 2-classes (i.e. positive and negative) performance. GRU performs worse than PV and MC with small and medium-scale data sets, whereas it outperforms both when trained on large-scale data. The outcome suggests that GRU memory units are beneficial for cross-domain, but require large-scale data in order to learn accurate word relationships. When tuned with samples of target data, GRU achieves accuracy comparable with the other methods with small and medium-scale data as well, proving that fine-tuning helps transfer learning across domains.

The rest of the paper is organized as follows. Section 2 reviews the literature about transfer learning, cross-domain sentiment classification and deep learning. The main features of the methods compared are outlined in Sect. 3. Section 4 describes, shows and discusses the experiments performed. Finally, Sect. 5 draws conclusions and paves the way for future work.

2 Related Work

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 [18], 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 [19], 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 are included [20]. Blitzer et al. discovered a measure of domain similarity supporting domain adaptation [21]. 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 [22]. 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 [23]. 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 [24]. 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 [25]. 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 [26]. 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 [27].

On the other hand, the advent of Deep Learning, whose a review can be found in [28], brought to a dramatic improvement in sentiment classification. Socher et al. introduced the Recursive Neural Tensor Networks to foster single sentence sentiment classification [11]. 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 [29]. 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 [30]. The ability of DMN 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 [11]. 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 [14]. 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 [13]. Wang et al. combined Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for sentiment analysis of short texts, taking advantage of the coarse-grained local features generated by CNN and long-distance dependencies learnt via RNN [31]. Chen et al. proposed a three-steps approach to learn a sentiment classifier for product reviews. First, they learnt a distributed representation of each review by a one-dimensional CNN. Then, they employed a RNN with gated recurrent units to learn distributed representations of users and products. Finally, they learnt a sentiment classifier from user, product and review representations [32].

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 [33] to extract domain-independent features in an unsupervised fashion, which can help transferring the knowledge extracted from a source domain to a target domain [34]. 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 Description

This Section firstly outlines the features of the methods used for the investigation. Then fine-tuning is described, along with the reason why it can be beneficial for transfer learning and cross-domain sentiment classification. The techniques compared in our previous work [1] were Paragraph Vector (referred as PV hereinafter), proposed in [12], and a Markov Chain (referred as MC hereinafter) based algorithm introduced in [15] and extended in [16], whereas Gated Recurrent Unit (GRU) [17] is added to the investigation in this work.

Careful readers can find further details on the approaches described below in [12,15–17].

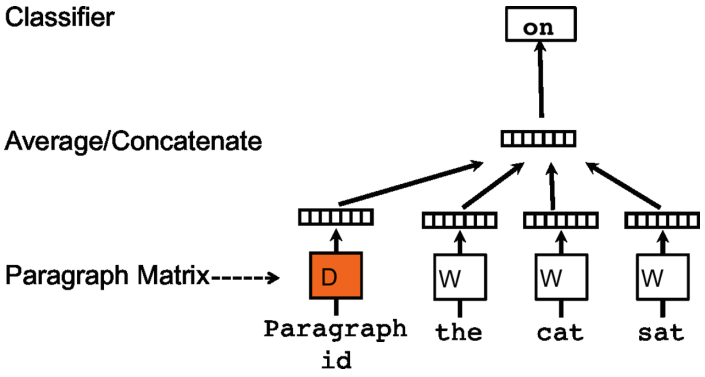

Fig. 1. The figure [12] shows a framework for learning the Distributed Memory Model of Paragraph Vector (PV-DM). With respect to word vectors, an additional paragraph token is mapped to a vector via matrix D. In this model, the concatenation or average of this vector with a context of three words is 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.

3.1 Paragraph Vector

PV 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 chunks of text, such as sentences, paragraphs, and documents. However, bag-of-words features lose the ordering of the words and 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 a 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 Fig. 1, is called the Distributed Memory Model of Paragraph Vector (PV-DM).

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 Paragraph Vector, shown in Fig. 2, is called the Distributed Bag of Words version (PV-DBOW).

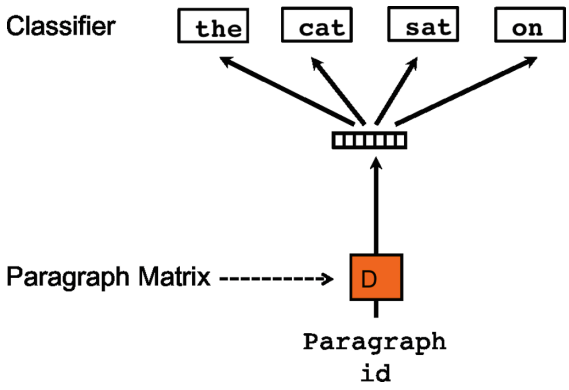

Fig. 2. The figure [12] shows the Distributed Bag of Words version of Paragraph Vector (PV-DBOW). The paragraph vector is trained to predict the words in a small window.

Both word vectors and paragraph vectors are trained by means of the stochastic gradient descent and backpropagation [35].

Sentiment classification requires sequential data to be handled, because document semantics is typically affected by 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 chunks of text, 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 [12] 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 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 PV able to automatically manage transfer learning, once enough data are available for training, as shown in [1].

As described in [12], 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.

3.2 Markov Chain

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 will be 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 will 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 (Fig. 3). The information flow is possible by exploiting the edges in the Markov Chain that, as previously stated, represent term-term relationships.

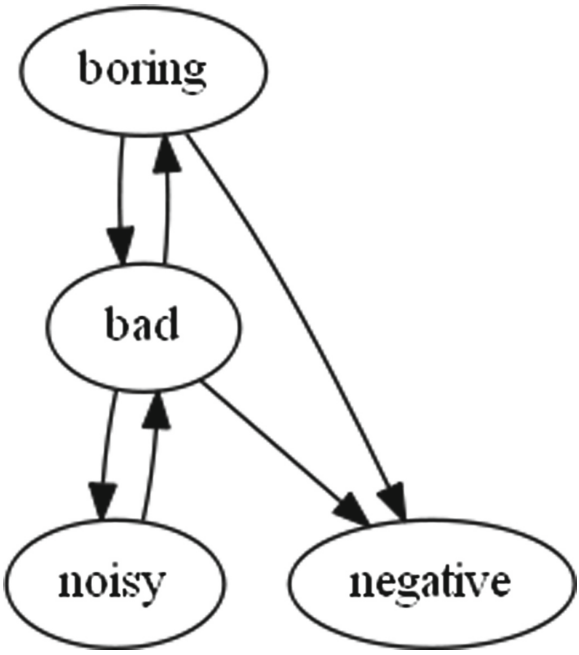

Fig. 3. The figure [15] shows transfer learning in Markov Chain from a book specific term like *boring* to an electrical appliance specific term like *noisy* through a common term like *bad*.

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, Markov Chain before performs 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.

3.3 Gated Recurrent Unit

Gated Recurrent Unit (GRU), proposed by Cho et al. [17], is an evolution of Long Short-Term Memory (LSTM), presented by Hochreiter and Schmidhuber [36]. LSTM is a deep architecture that has been introduced to overcome the vanishing (or blowing up) gradient problem [37] that affects recurrent nets when signals are backpropagated through long time sequences. Indeed, LSTM can learn to bridge time intervals in excess of 1000 discrete time steps without loss of short time lag

capabilities, by enforcing constant error flow through internal states of special units. LSTM units are memory cells composed of different gates, namely input gate, output gate, and forget gate. The input gate of a unit u allows protecting the memory contents stored in j from perturbation by irrelevant inputs. The output gate of u allows protecting other units from perturbation by irrelevant memory contents stored in j. The forget gate of u allows forgetting the memory contents that are no longer relevant. An LSTM unit is able to decide whether to keep the existing memory content via the gates. Basically, if the LSTM unit detects a relevant feature from an input sequence, it is able to preserve this information over a long distance. This property makes LSTM suitable for capturing long-distance dependencies.

GRU extends LSTM by making each recurrent unit adaptively capture dependencies of different time scales. A GRU is similar to the LSTM unit, but it only presents two gates, as shown in Fig. 4. The activation of the GRU is ruled by an update gate, which controls how much information from the previous hidden state will carry over to the current hidden state. A reset gate allows the hidden state to drop any information that is found to be irrelevant later in the future. As each hidden unit has separate reset and update gates, it will learn to capture dependencies over different time scales.

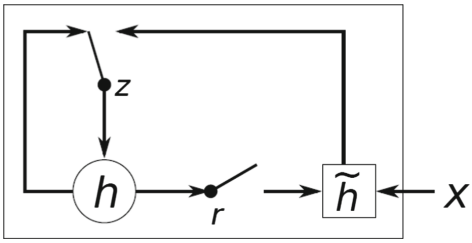

Fig. 4. The figure [17] shows a GRU. The update gate z selects whether the hidden state is to be updated with a new hidden state $\widetilde{h}$. The reset gate r decides whether the hidden state is ignored.

LSTM based schemes have already been proved to work well in sentiment classification [14]. In this work, GRU is applied to cross-domain sentiment classification, to assess whether it is able to automatically bridge the semantic gap between the source and the target domain. Alike PV, GRU is a deep architecture and does not rely on a transfer learning mechanism. However, GRU gates, which allow each unit working as a memory wherein relevant information can be stored and preserved, make GRU suitable for cross-domain problems. Important domain-independent information can be automatically extracted by GRU if trained with an appropriate amount of data.

3.4 Fine-Tuning in Cross-domain

Fine-tuning consists in using a labelled sample of target instances to refine a model previously learnt on the source domain. The sample should be reasonably small for both theoretical and practical reasons. From a theoretical point of view, the cross-domain task would be converted into an in-domain problem if the sample used for fine-tuning was too large. Moreover, cross-domain would be no longer needed if an appropriate amount of labelled target instances were available. An in-domain model could be easily learnt in that case. On the other hand, readers already know that cross-domain learning is essential from a practical point of view, since most real-world data are unlabelled. Finding a large labelled sample is challenging in practice, and manually labelling it is even infeasible. Therefore, using a large sample would not be a viable alternative for tuning a pre-trained model. On the other hand, if the sample was small, categorisation by human experts would become a good option to increase cross-domain efficacy.

Beyond being a good trade-off between its cost and the improvement of performance that guarantees, fine-tuning could be even critical for techniques that do not rely on explicit transfer learning mechanisms. In particular, this work assesses whether fine-tuning of deep neural networks can bring to an improvement of ad-hoc cross-domain solutions.

4 Experiments

This Section shows some experiments to assess whether GRU, alike PV, is automatically able to handle language heterogeneity in cross-domain tasks, in relation to the amount of training data available. The effect of fine-tuning on deep architectures is also discussed, showing that it can help improving cross-domain performance, especially with small-scale data sets. Markov Chain has been implemented in a custom Java-based framework. Paragraph Vector relies on 0.12.4 *gensim* release [38], a Python-based open sourced and freely available framework[1]. For Gated Recurrent Unit (GRU) we used the Python-based implementation provided by Keras[2], choosing TensorFlow as back-end.

4.1 Setup

A common benchmark text set has been used to compare results with our previous work [1], that is 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 (i.e. very negative) to 5 (i.e. very positive). The reviews whose scores were 1 and 2 have been mapped to the negative category, those whose scores were 4 and 5 to the positive one, whereas we discarded those whose score was 3 because they were likely to

[1] http://nlp.fi.muni.cz/projekty/gensim/.
[2] https://keras.io/layers/recurrent/.
[3] http://jmcauley.ucsd.edu/data/amazon/.

express a neutral sentiment orientation. To assess to what extent the amount of training data affects performance, source-target partitions with three orders of magnitude have been tested, preserving 80%–20% as source-target ratio, and balancing positive and negative examples. The small-scale data set has 1600 instances as the training set and 400 as the test set; the medium-scale 16000 and 4000 respectively; and the large-scale 80000 and 20000 respectively. Accuracy (i.e. the percentage of correctly classified instances) has been measured for each source-target configuration, averaging results on 10 different training-test partitions to reduce the variance, that is, the sensitivity to small fluctuations in the training set.

The same configurations of our previous work [1] have been used for Paragraph Vector and Markov Chain. The Distributed Bag of Words version (PV-DBOW) [12] has been chosen for PV, selecting 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. The initial learning rate has been set to 0.025, letting it linearly decade to 0.001 in 30 epochs. Readers can refer to [12,39] for details on the parameters. A logistic classifier, whose regression coefficients have been estimated through the Newton-Raphson method, has been used to perform sentiment classification.

In conformity with the previous work [1], we relied on the Markov Chain algorithm introduced in [15]. The relative frequency of terms in documents has been chosen as the term weighting measure [40]. Feature selection by means of χ^2 scoring function has been carried out to mitigate the curse of dimensionality that inherently affects dense bag-of-words models. 750, 10000 and 25000 terms have been chosen for the small-scale, medium-scale, and large-scale data sets respectively. Readers can refer to [15,16] for further details on the method.

For the GRU-based architecture, 3 main layers have been chosen: the first two are GRU layers, and the last one is a dense layer, fully-connected to the classes. Each GRU layer consists of 128 units as the output space dimensionality, whereas Glorot uniform initialisation [41] has been performed for the kernel weights matrix. 10% of the inputs to the second GRU layer have been discarded via dropout, in order to improve network robustness to noise. Adam optimizer [42] has been used to perform stochastic gradient descent, with binary cross-entropy as the loss function to optimize. Default values have been kept for the other parameters. Readers can refer to Keras documentation for further details.

The analysis below mainly focuses on cross-domain sentiment classification, where transfer learning is typically required to bridge the semantic gap between distinct domains. In-domain experiments have been shown just to have a baseline for the cross-domain comparison between GRU and the techniques already examined in the previous work [1]. The impact of fine-tuning on the deep architectures is finally addressed, assessing whether tuning allows increasing their cross-domain performance, since PV and GRU do not provide explicit transfer learning mechanisms.

4.2 In-domain Experiments

In-domain results are presented for a matter of comparison with the previous work [1]. They act as a baseline for cross-domain comparison. Table 1 shows the results over the 4 domains of the Amazon reviews dataset, namely Books (B), Movies (M), Electronics (E) and Clothing-Shoes-Jewelry (J).

Table 1. In-domain comparison among the three techniques used in this paper. 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. $X \rightarrow X$ means that the model has been learnt on reviews from a domain X and then applied to different reviews from the same domain. Values have been rounded to one decimal place for space reason.

Domain(s)	1.6k-0.4k			16k-4k			80k-20k		
	PV	*MC*	*GRU*	*PV*	*MC*	*GRU*	*PV*	*MC*	*GRU*
In-domain experiments									
$B \rightarrow B$	67.3%	79.3%	83.5%	75.4%	81.9%	73.8%	84.7%	83.8%	89.6%
$M \rightarrow M$	79.8%	91.2%	76.8%	74.9%	82.4%	78.9%	84.1%	80.2%	79.5%
$E \rightarrow E$	79.3%	92.0%	80.8%	80.2%	80.7%	82.5%	85.6%	84.4%	85.2%
$J \rightarrow J$	75.5%	72.0%	82.3%	80.1%	83.8%	85.0%	85.3%	87.0%	84.4%
Average	**75.4%**	**83.6%**	**80.8%**	**77.6%**	**82.2%**	**80.0%**	**84.9%**	**83.9%**	**84.7%**

GRU achieves performance comparable with the other techniques. The outcome is not surprising for many reasons. Firstly, GRU has a recurrent architecture, suitable for modelling sequences of terms. Secondly, GRU is able to capture dependencies of different time scales. Relationships among terms arise independently of how much they are distant. Finally, GRU can store relevant information through time, working as a memory. Readers can find further discussion on PV and MC in the previous paper [1].

4.3 Cross-domain Experiments

The following experiment aims to compare the performance of GRU with PV and MC in cross-domain sentiment classification. The goal is to assess whether the memory mechanism of GRU, which allows preserving relevant information through time, makes it suitable for cross-domain learning. This comparison also strengthens our earlier investigation [1] on deep learning in cross-domain sentiment classification.

The analysis involves all source-target configurations of the four domains, namely $B \rightarrow E$, $B \rightarrow M$, $B \rightarrow J$, $E \rightarrow B$, $E \rightarrow M$, $E \rightarrow J$, $M \rightarrow B$, $M \rightarrow E$, $M \rightarrow J$, $J \rightarrow B$, $J \rightarrow E$, $J \rightarrow M$. The detailed results are shown in Table 2, whereas the average trend across domains is represented in Fig. 5.

Table 2. Comparison between GRU, PV and MC in cross-domain sentiment classification. 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. Values have been rounded to one decimal place for space reason.

Domain(s)	1.6k-0.4k			16k-4k			80k-20k		
	PV	*MC*	*GRU*	*PV*	*MC*	*GRU*	*PV*	*MC*	*GRU*
Cross-domain experiments (*source* → *target*)									
$B \rightarrow E$	70.8%	69.3%	64.0%	67.3%	71.2%	73.2%	73.2%	74.1%	78.2%
$B \rightarrow M$	66.8%	70.9%	65.0%	80.3%	79.3%	77.8%	82.0%	79.0%	83.3%
$B \rightarrow J$	73.3%	79.7%	69.5%	70.6%	71.8%	78.0%	74.9%	76.0%	82.7%
$E \rightarrow B$	74.0%	54.0%	65.8%	78.8%	80.1%	69.5%	76.9%	79.2%	77.0%
$E \rightarrow M$	71.5%	56.8%	64.7%	76.2%	76.2%	73.4%	76.9%	77.2%	79.3%
$E \rightarrow J$	82.8%	74.3%	69.2%	79.5%	80.5%	82.0%	80.8%	81.9%	85.7%
$M \rightarrow B$	74.8%	65.8%	59.0%	85.6%	86.1%	71.7%	85.2%	83.8%	81.2%
$M \rightarrow E$	71.8%	68.2%	69.5%	75.3%	77.1%	74.7%	74.8%	72.9%	79.5%
$M \rightarrow J$	82.3%	82.0%	68.0%	73.5%	74.9%	77.0%	77.0%	78.6%	82.3%
$J \rightarrow B$	66.3%	75.3%	60.5%	69.6%	80.6%	68.2%	76.5%	78.6%	77.2%
$J \rightarrow E$	76.5%	80.6%	77.0%	78.6%	79.8%	78.4%	80.1%	81.8%	82.2%
$J \rightarrow M$	74.3%	81.3%	63.5%	70.8%	74.3%	72.7%	76.1%	77.9%	77.6%
Average	**73.7%**	**71.5%**	**66.3%**	**75.5%**	**77.7%**	**74.7%**	**77.9%**	**78.4%**	**80.5%**

The average trend is pretty clear: the more data GRU relies on, the more it performs well. Indeed, GRU underperforms the other techniques with small-scale data. Increasing the number of training instances, GRU experienced a dramatic growth in accuracy, becoming comparable with both PV and MC with medium-scale data and even outperforming them with large-scale data. A reasonable explanation for this behaviour is that the memory mechanism of GRU needs an appropriate amount of data in order to learn what is actually relevant within a review. Somebody might argue that, looking at in-domain results in Table 1, GRU achieves good performance even with small data sets. This means that GRU needs few data to capture intra-domain term relationships, whereas few facts are not enough to capture inter-domain dependencies in absence of some explicit transfer learning mechanism. This is rational. Just think that in a single domain identifying the polarity-bearing terms could be enough to understand the overall sentiment orientation of the review, whereas the same does not hold between distinct domains, because of language heterogeneity. The polarity-bearing terms of the source domain generally differ from those of the target domain. In order to support the knowledge transfer from source to target, cross-domain makes it necessary to identify relevant hidden concepts rather than important terms. Careful readers could argue that PV, similarly to GRU,

does not provide for a transfer learning phase, achieving good performance with small-scale data anyway. This is true, but it should not be forgotten that PV is able to capture word semantics without supervision [12]. This feature makes PV suitable for bridging the inter-domain semantic gap, as shown in [1].

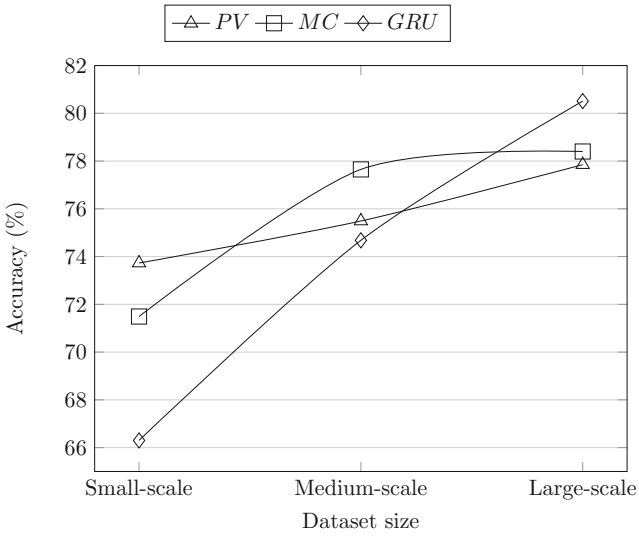

Fig. 5. Average accuracy achieved by the compared methods in cross-domain sentiment classification. The number of instances respectively used for small-scale, medium-scale and large-scale are reported in Sect. 4.1.

The outcome of cross-domain experiments suggests that gated recurrent units are automatically able to decide which information is better to preserve even across heterogeneous domains. However, GRU needs a large-scale training set in order to perform well in cross-domain tasks. Since GRU does not rely on explicit transfer learning mechanisms, it requires more data in order to extract hidden relevant concepts to bridge the semantic gap between distinct domains.

4.4 Experiments with Fine-Tuning

The experiments illustrated below assess the effectiveness of fine-tuning in supporting deep learning techniques in cross-domain sentiment classification. As explained in Sect. 3.4, fine-tuning of a pre-trained model can be useful in practice only if the labelled sample of the target domain is reasonably small. If it was too large, cross-domain would lose its benefits and, at the same time, in-domain approaches would be both feasible and preferable. For this purpose, 250 and 500 target examples have been used to assess the potentiality of fine-tuning as transfer learning mechanism. The detailed cross-domain results with fine-tuning

are shown in Table 3, whereas the average trend is plotted in Fig. 6 and compared with the accuracy that PV and GRU obtained without tuning on target instances.

Table 3. Comparison between GRU, PV and MC in cross-domain sentiment classification with fine-tuning on a small set of target instances. 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. 250 and 500 target instances have been used for tuning.

Domain(s)	1.6k-0.4k		16k-4k		80k-20k	
	PV	*GRU*	*PV*	*GRU*	*PV*	*GRU*
Fine-tuning with 250 instances (*source* → *target*)						
$B \rightarrow E$	70.56%	69.00%	73.86%	74.85%	72.40%	79.05%
$B \rightarrow M$	67.78%	68.50%	76.33%	79.45%	83.83%	84.51%
$B \rightarrow J$	76.67%	71.50%	71.03%	79.80%	75.16%	82.03%
$E \rightarrow B$	63.89%	68.50%	78.61%	70.20%	74.73%	77.83%
$E \rightarrow M$	73.78%	67.00%	65.28%	73.85%	79.18%	80.17%
$E \rightarrow J$	82.22%	71.00%	79.14%	84.60%	81.14%	86.44%
$M \rightarrow B$	79.17%	67.50%	80.08%	72.95%	81.22%	81.22%
$M \rightarrow E$	73.89%	72.50%	77.33%	77.10%	74.38%	79.70%
$M \rightarrow J$	82.50%	75.50%	75.58%	78.85%	77.92%	82.83%
$J \rightarrow B$	64.17%	67.50%	74.81%	70.30%	75.22%	78.78%
$J \rightarrow E$	73.89%	77.50%	83.11%	78.50%	80.70%	82.51%
$J \rightarrow M$	70.83%	68.00%	62.53%	75.25%	78.25%	79.19%
Average	**73.28%**	**70.33%**	**74.81%**	**76.31%**	**77.84%**	**81.19%**
Fine-tuning with 500 instances (*source* → *target*)						
$B \rightarrow E$	70.72%	70.50%	73.54%	73.95%	72.24%	78.56%
$B \rightarrow M$	67.56%	68.00%	76.28%	79.55%	83.66%	84.07%
$B \rightarrow J$	76.17%	78.00%	70.53%	80.50%	74.89%	82.30%
$E \rightarrow B$	66.83%	69.00%	78.21%	71.85%	74.97%	77.39%
$E \rightarrow M$	73.03%	72.50%	64.99%	76.65%	79.23%	80.82%
$E \rightarrow J$	81.33%	76.50%	79.02%	82.15%	81.15%	85.19%
$M \rightarrow B$	79.33%	69.00%	80.94%	76.55%	81.35%	81.78%
$M \rightarrow E$	74.61%	72.50%	77.55%	77.25%	74.34%	79.91%
$M \rightarrow J$	84.17%	77.50%	74.65%	79.25%	77.70%	82.33%
$J \rightarrow B$	66.06%	70.00%	74.42%	71.85%	75.62%	78.52%
$J \rightarrow E$	74.83%	78.00%	82.84%	79.55%	80.28%	83.66%
$J \rightarrow M$	71.66%	68.50%	62.33%	76.60%	78.04%	79.59%
Average	**73.86%**	**72.50%**	**74.61%**	**77.14%**	**77.79%**	**81.18%**

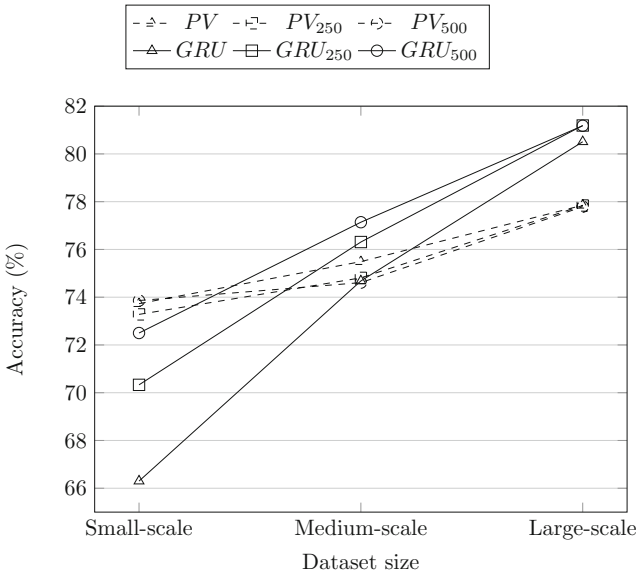

Fig. 6. Average accuracy achieved by the compared methods when fine-tuning on small samples of target instances is performed to foster cross-domain sentiment classification. The number of instances respectively used for small-scale, medium-scale and large-scale are reported in Sect. 4.1. The subscripts 250 and 500 are referred to the number of instances sampled from the target domain in order to perform fine-tuning.

The first outcome that catches the eye is that PV is almost unaffected by fine-tuning, regardless of the data set size. This behaviour is explained by the ability of PV to capture word semantics without supervision. PV automatically handles language heterogeneity by discovering hidden relationships between semantically similar words [12]. On the other hand, the benefits of fine-tuning dramatically affect GRU, which is not inherently able to align domains without supervision. The improvement is particularly evident with small-scale data, and decreases by growing the amount of source data employed to pre-train the model. The reason is pretty obvious. When few training data are available, GRU cannot capture inter-domain dependencies, and even a small sample of target data leads to a significant boost of performance. The impact is a bit reduced with medium-scale data, mainly for two factors. The first factor is the increased capability of GRU in bridging the inter-domain semantic gap without fine-tuning, as already shown in Sect. 4.3. The second factor is that 250 and 500 instances are two orders of magnitude less than the dataset size considered, whereas the small-scale data were just one order of magnitude more than the amount used for tuning. It is obviously challenging to increase the performance of a model pre-trained on a set of medium-scale data, by using only such a small sample for tuning. The same two factors also affect performance improvement when large-scale data are taken into account.

Despite the few instances used, fine-tuning is beneficial to GRU on average in all the considered configurations. With small-scale data, the sample of 250 target instances improves accuracy by approximately 4%, whereas doubling the tuning instances, accuracy increases by about 2% more. With medium-scale data, the smaller sample boosts accuracy by about 1.6%, whereas the bigger one by less than 1% with respect to the smaller. Finally, accuracy increases by less than 1% with respect to the configuration without tuning when large-scale data are considered, independently of the size of the tuning sample.

The outcome of the analysis proves that fine-tuning on a small sample of labelled target data is beneficial to deep architectures that do not have neither explicit transfer learning mechanisms nor the capability of automatically detecting semantically similar terms without supervision.

5 Conclusions

In this work, the investigation on deep learning in cross-domain sentiment classification, started in [1], has been carried on.

A Gated Recurrent Unit based architecture has been added to the previous comparison, which already took into account Paragraph Vector, an unsupervised deep learning technique not designed for cross-domain purposes, and a Markov Chain based method tailored to transfer learning and cross-domain sentiment classification. Moreover, fine-tuning of a pre-trained model has been attempted to assess its impact on cross-domain as explicit transfer learning mechanism. The model pre-trained on the source domain was tuned on a small sample of labelled target instances. The sample should be small in order for human experts to manually label data without too much effort. Moreover, if a large amount of labelled data was available, in-domain approaches would be preferable, as they are generally more effective than cross-domain ones.

The cross-domain experiments without fine-tuning show that GRU needs many instances in order to learn bridging the semantic gap between the source and the target domain. Indeed, GRU performs poorly with small-scale data (e.g. 2000 examples), achieves accuracy comparable with the other techniques with medium-scale data (e.g. 20000 examples), and even outperforms both with large-scale data (e.g. 100000). The outcome also means that, once enough data are available for training, GRU is able to bridge the inter-domain semantic gap without explicit transfer learning mechanisms. This ability is supposedly due to GRU gates, which allow each unit working as a memory wherein relevant information can be stored and preserved.

The deep architectures analysed manifest different behaviours in the experiments with fine-tuning. PV does not take advantage of fine-tuning, since it is able to capture word semantics as well as word relationships without supervision. On the other hand, fine-tuning is beneficial to GRU, because it acts as a transfer learning mechanism. The less training examples have been used to pre-train the model on the source domain, the higher impact fine-tuning has had on performance. As expected, a greater amount of tuning data (e.g. 500 reviews

rather than 250) brings to better performance with small-scale data. The impact of this factor decreases by augmenting the dataset cardinality, and completely vanishes with large-scale data.

The analysis carried out in this work confirms that deep architectures are promising for cross-domain sentiment classification, although the techniques used in this investigation do not explicitly incorporate transfer learning mechanisms. Some features make deep nets suitable for bridging the inter-domain semantic gap, like the capability of PV to learn word semantics and relationships without supervision, and the memory mechanism of GRU that allows preserving relevant information through time. When combined with explicit transfer learning mechanisms as fine-tuning, deep learning techniques achieve accuracy comparable with or better than ad-hoc cross-domain solutions. Moreover, the fact that deep learning algorithms are able to take advantage of large-scale data is extremely important in nowadays big data scenarios, where scalability always is a requirement.

Future work will focus on combining different deep learning approaches, in order to take advantage of the respective benefits. We argue that this study is a start point to overcome ad-hoc solutions for cross-domain sentiment classification. A possibility is to combine deep approaches to learn semantic-bearing word representation - like Paragraph Vector, Glove [43], ELMo [44], etc. - with deep architectures with some memory mechanism, such as Gated Recurrent Unit, Differentiable Neural Computer [45], Dynamic Memory Network, etc. (see in [46] for an extensive treatment in transfer learning). Moreover this study can be extended to cope with other emerging text classification problems where large data sets are unlabelled, such as in thread of conversational messages of social networks and discussion forums [47,48].

References

1. Domeniconi, G., Moro, G., Pagliarani, A., Pasolini, R.: On deep learning in cross-domain sentiment classification. In: Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management: KDIR, INSTICC, vol. 1, pp. 50–60. SciTePress (2017)
2. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C., Zhai, C. (eds.) Mining Text Data, pp. 415–463. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-3223-4_13
3. Domeniconi, G., Moro, G., Pagliarani, A., Pasolini, R.: Learning to predict the stock market Dow Jones index detecting and mining relevant tweets. In: Fred, A.L.N., Filipe, J. (eds.) Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Funchal, Madeira, Portugal, 1–3 November 2017, vol. 1, pp. 165–172. SciTePress (2017)
4. Domeniconi, G., Moro, G., Pagliarani, A., Pasini, K., Pasolini, R.: Job recommendation from semantic similarity of Linkedin users' skills. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods: ICPRAM, INSTICC, vol. 1, pp. 270–277. SciTePress (2016)

5. Lena, P.D., Domeniconi, G., Margara, L., Moro, G.: GOTA: GO term annotation of biomedical literature. BMC Bioinform. **16**, 346 (2015)
6. Domeniconi, G., Moro, G., Pasolini, R., Sartori, C.: Iterative refining of category profiles for nearest centroid cross-domain text classification. In: Fred, A., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) IC3K 2014. CCIS, vol. 553, pp. 50–67. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25840-9_4
7. Shrivastava, A., Malisiewicz, T., Gupta, A., Efros, A.A.: Data-driven visual similarity for cross-domain image matching. ACM Trans. Graph. **30**, 154:1–154:10 (2011)
8. Domeniconi, G., Masseroli, M., Moro, G., Pinoli, P.: Cross-organism learning method to discover new gene functionalities. Comput. Meth. Progr. Biomed. **126**, 20–34 (2016)
9. Domeniconi, G., Masseroli, M., Moro, G., Pinoli, P.: Random perturbations of term weighted gene ontology annotations for discovering gene unknown functionalities. In: Fred, A., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) IC3K 2014. CCIS, vol. 553, pp. 181–197. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25840-9_12
10. Domeniconi, G., Masseroli, M., Moro, G., Pinoli, P.: Discovering new gene functionalities from random perturbations of known gene ontological annotations. In: KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Rome, Italy, 21–24 October 2014, pp. 107–116. SciTePress (2014)
11. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment TreeBank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642. Association for Computational Linguistics, Stroudsburg (2013)
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning, ICML 2014, vol. 32, pp. II-1188–II-1196. JMLR.org (2014)
13. Zhang, X., LeCun, Y.: Text understanding from scratch. CoRR abs/1502.01710 (2015)
14. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: EMNLP, pp. 1422–1432. The Association for Computational Linguistics (2015)
15. Domeniconi, G., Moro, G., Pagliarani, A., Pasolini, R.: Markov chain based method for in-domain and cross-domain sentiment classification. In: Fred, A.L.N., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) KDIR 2015 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Lisbon, Portugal, 12–14 November 2015, vol. 1, pp. 127–137. SciTePress (2015)
16. Domeniconi, G., Moro, G., Pagliarani, A., Pasolini, R.: Cross-domain sentiment classification via polarity-driven state transitions in a Markov model. In: Fred, A., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) IC3K 2015. CCIS, vol. 631, pp. 118–138. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-52758-1_8
17. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1724–1734. Association for Computational Linguistics (2014)
18. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. J. Artif. Intell. Res. **26**, 101–126 (2006)

19. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**, 1345–1359 (2010)
20. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: a case study. In: Proceedings of Recent Advances in Natural Language Processing (RANLP) (2005)
21. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Carroll, J.A., van den Bosch, A., Zaenen, A. (eds.) ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, 23–30 June 2007, pp. 440–447. The Association for Computational Linguistics (2007)
22. Pan, S.J., Ni, X., Sun, J., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Rappa, M., Jones, P., Freire, J., Chakrabarti, S. (eds.) Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, 26–30 April 2010, pp. 751–760. ACM (2010)
23. He, Y., Lin, C., Alani, H.: Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In: Lin, D., Matsumoto, Y., Mihalcea, R. (eds.) The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19–24 June 2011, Portland, Oregon, USA, pp. 123–131. The Association for Computer Linguistics (2011)
24. Bollegala, D., Weir, D.J., Carroll, J.A.: Cross-domain sentiment classification using a sentiment sensitive thesaurus. IEEE Trans. Knowl. Data Eng. **25**, 1719–1731 (2013)
25. Zhang, Y., Hu, X., Li, P., Li, L., Wu, X.: Cross-domain sentiment classification-feature divergence, polarity divergence or both? Pattern Recogn. Lett. **65**, 44–50 (2015)
26. Franco-Salvador, M., Cruz, F.L., Troyano, J.A., Rosso, P.: Cross-domain polarity classification using a knowledge-enhanced meta-classifier. Knowl.-Based Syst. **86**, 46–56 (2015)
27. Bollegala, D., Mu, T., Goulermas, J.Y.: Cross-domain sentiment classification using sentiment sensitive embeddings. IEEE Trans. Knowl. Data Eng. **28**, 398–410 (2016)
28. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**, 436–444 (2015)
29. dos Santos, C.N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Hajic, J., Tsujii, J. (eds.) COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 23–29 August 2014, Dublin, Ireland, pp. 69–78. ACL (2014)
30. Kumar, A., et al.: Ask me anything: dynamic memory networks for natural language processing. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, 19–24 June 2016. JMLR Workshop and Conference Proceedings, vol. 48, pp. 1378–1387. JMLR.org (2016)
31. Wang, X., Jiang, W., Luo, Z.: Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.) COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016, pp. 2428–2437. ACL (2016)
32. Chen, T., Xu, R., He, Y., Xia, Y., Wang, X.: Learning user and product distributed representations using a sequence model for sentiment analysis. IEEE Comp. Int. Mag. **11**, 34–44 (2016)

33. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
34. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: a deep learning approach. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 28 June–2 July 2011, pp. 513–520. Omnipress (2011)
35. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
36. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
37. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **6**, 107–116 (1998)
38. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, pp. 45–50. ELRA (2010). http://is.muni.cz/publication/884893/en
39. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems: 27th Annual Conference on Neural Information Processing Systems 2013, 5–8 December 2013, Lake Tahoe, Nevada, United States, vo. 26, pp. 3111–3119 (2013)
40. Domeniconi, G., Moro, G., Pasolini, R., Sartori, C.: A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf.idf. In: Helfert, M., Holzinger, A., Belo, O., Francalanci, C. (eds.) DATA 2015. CCIS, vol. 584, pp. 39–58. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30162-4_4
41. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, D.M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Proceedings, vol. 9, pp. 249–256. JMLR.org (2010)
42. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014)
43. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1532–1543. ACL (2014)
44. Peters, M.E., et al.: Deep contextualized word representations. In: Walker, M.A., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018 (Long Papers), New Orleans, Louisiana, USA, 1–6 June 2018, vol. 1, pp. 2227–2237. Association for Computational Linguistics (2018)
45. Graves, A., et al.: Hybrid computing using a neural network with dynamic external memory. Nature **538**, 471–476 (2016)

46. Moro, G., Pagliarani, A., Pasolini, R., Sartori, C.: Cross-domain & in-domain sentiment analysis with memory-based deep neural networks. In: Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management: KDIR, INSTICC, vol. 1. SciTePress (2018)
47. Domeniconi, G., Semertzidis, K., Moro, G., Lopez, V., Kotoulas, S., Daly, E.M.: Identifying conversational message threads by integrating classification and data clustering. In: Francalanci, C., Helfert, M. (eds.) DATA 2016. CCIS, vol. 737, pp. 25–46. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62911-7_2
48. Domeniconi, G., Semertzidis, K., López, V., Daly, E.M., Kotoulas, S., Moro, G.: A novel method for unsupervised and supervised conversational message thread detection. In: DATA, pp. 43–54. SciTePress (2016)