Markov Chain based Method for In-Domain and Cross-Domain Sentiment Classification

Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani and Roberto Pasolini
DISI, Università degli Studi di Bologna, Via Venezia 52, Cesena, Italy

Keywords: Transfer Learning, Sentiment Classification, Markov Chain, Parameter Tuning, Language Independence, Opinion Mining.

Abstract: Sentiment classification of textual opinions in positive, negative or neutral polarity, is a method to understand people thoughts about products, services, persons, organisations, and so on. Interpreting and labelling opportunely text data polarity is a costly activity if performed by human experts. To cut this labelling cost, new *cross domain* approaches have been developed where the goal is to automatically classify the polarity of an unlabelled *target text set* of a given domain, for example movie reviews, from a labelled *source text set* of another domain, such as book reviews. Language heterogeneity between source and target domain is the trickiest issue in cross-domain setting so that a preliminary transfer learning phase is generally required. The best performing techniques addressing this point are generally complex and require onerous parameter tuning each time a new source-target couple is involved. This paper introduces a simpler method based on the Markov chain theory to accomplish both transfer learning and sentiment classification tasks. In fact, this straightforward technique requires a lower parameter calibration effort. Experiments on popular text sets show that our approach achieves performance comparable with other works.

1 INTRODUCTION

Text classification copes with the problem of automatically organising a corpus of documents into a predefined set of *categories* or *classes*, which usually are the document topics, like for instance sport, politics, cinema and so on. Differently, *sentiment classification* is a particular text classification task that organises documents according to their polarity, such as positive, negative and possibly neutral.

Any supervised approach learns a classification model from a training set of documents, labelled according to their topics, in order to classify new unlabelled documents into the same set of topics. The more new documents reflect the peculiarity of training set, the more the classification is accurate. The accuracy of the classification model is evaluated by applying the model to a labelled test set. Generally new and possibly better classification models of the test set are extracted from the training set by varying the parameters of the learning algorithm adopted. The classical approach of sentiment classification assumes that both training set and test set deal with the same topic. This modus operandi, known as *in-domain* sentiment classification results in optimal effectiveness, being training set and test set lexically and semantically similar.

However this approach is inapplicable when the text set we want to classify is completely unlabelled, which is the most frequent real case, such as with tweets, blog opinions or with comments in social networks. Therefore, having a document set treating a certain topic, for instance book reviews with each review labelled according to its polarity, it is desirable to extract from this document set a classification model capable of classifying the polarity of new document sets whatever kind of topic they deal with, for instance electrical appliance reviews: this approach is called *cross-domain* sentiment classification. Since classification is typically driven by terms, the most critical issue in cross-domain setting is language heterogeneity, whereas just the classes are usually the same (i.e. positive, negative or neutral). For instance, if we consider two different domains like books and electrical appliances, we may notice that a book can be interesting, boring, funny, but the same attributes are meaningless in describing electrical appliances.

This work was partially supported by the project "GenData 2020", funded by the Italian MIUR.

Domeniconi, G., Moro, G., Pagliarani, A. and Pasolini, R..
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 (IC3K 2015) - Volume 1: KDIR*, pages 127-137
ISBN: 978-989-758-158-8

On the other hand, an electrical appliance can be efficient, noisy, clean, but again these attributes are not the most relevant in book reviews.

Transfer learning approaches are employed in cross-domain setting to overcome the lexical gap between source and target text sets. In literature, several methods have been proposed with the aim of either adapting the representation of source domain to fit the target domain or mapping both domains in a common latent space (Dai et al., 2007; Xue et al., 2008; Li et al., 2012). Among them, the most popular are clustering algorithms and approaches that make use of feature expansion and external, sometimes hierarchical, knowledge bases. These techniques, employed in conjunction with standard text classification algorithms, lead to good results in sentiment classification (Tan et al., 2008; Qiu et al., 2009; Melville et al., 2009). Nevertheless, a heavy parameter tuning, which is needed to achieve accurate performance, is a bottleneck each time a new text set has to be classified, in fact the parameter values that yield optimal accuracy for a text set, usually do not produce analogous best results with different corpora. Therefore, defining algorithms that are not affected (or slightly affected) by the parameter tuning problem represents an open research challenge. (Domeniconi et al., 2014) propose a novel method to solve this issue in the text classification context; however sentiment classification is not taken into account in the study.

In this paper we introduce a novel method for in-domain and cross-domain sentiment classification based on Markov chain theory that performs both transfer learning from a source domain to a target domain and sentiment classification over target domain. The basic idea is to model semantic information looking at term distribution within a corpus. To accomplish this task, we represent the document corpus as a graph characterised by a node for each different term and a link between co-occurrent terms. In this way, we mold a semantic network where information can flow allowing transfer learning from source specific terms to target specific ones. Similarly, we also add categories to graph nodes in order to model semantic information between source specific terms and classes.

The semantic graph is implemented by using a Markov chain, whose states represent both terms and classes, modelling state transitions as a function of term(-class) co-occurrences in documents. In particular, Markov chain terms belong either to source domain or to target domain, whereas connections with class states are available only starting from terms that occur in source domain. This mathematical model is appropriate to spread semantic information throughout the network by performing steps (i.e. state transitions) in the Markov chain. The simultaneous existence of terms belonging to source domain, terms belonging to target domain and categories in the Markov chain ensures both the transfer learning capability and the classification capability, as will be argued in section 3. To the best of our knowledge, it is the first time that a Markov chain is used as a classifier in a sentiment classification task rather than for a support task such as part-of-speech (POS) tagging. Moreover, considering categories as Markov chain states is a novelty with reference to text classification as well, where the most common approach consists in building different Markov chains (i.e. one for each category) and evaluating the probability that a test document is being generated by each of them.

We performed experiments on publicly available benchmark text sets considering just 2 classes (i.e. positive and negative) and we tested performance in both in-domain and cross-domain sentiment classification, achieving an accuracy comparable with the best methods in literature. This means that our Markov chain based method succeeds in classifying document polarity. Moreover, we notice that less burdensome parameter tuning is required with respect to the state of the art. This peculiarity makes the approach particularly appealing in real contexts where both reduced computational costs and accuracy are essential.

The rest of the paper is organised as follows. Section 2 outlines the state of the art approaches about in-domain and cross-domain sentiment classification and Markov chains. Section 3 describes the novel method based on Markov chain theory. Section 4 illustrates the performed experiments, discusses the outcome and the comparison with other works. Lastly, section 5 summarises results and possible future works.

2 RELATED WORK

In cross-domain setting, transfer learning approaches are required to map a source domain to a target domain. More specifically, two transfer modes can be identified: *instance-transfer* and *feature-representation-transfer* (Pan and Yang, 2010). The former aims to bridge the inter-domain gap by adjusting instances from source to target. Vice versa, the latter pursues the same goal by mapping features of both source and target in a different space. In the text categorisation context, transfer learning has been fulfilled in some ways, for example by clustering together documents and words (Dai et al., 2007), by extending *probabilistic latent semantic analysis* also

to unlabelled instances (Xue et al., 2008), by extracting latent words and topics, both common and domain specific (Li et al., 2012), by iteratively refining target categories representation without a burdensome parameter tuning (Domeniconi et al., 2014; Domeniconi et al., 2015b).

Apart from the aforementioned, a number of different techniques have been developed solely for sentiment classification. For example, (Dave et al., 2003) draw on information retrieval methods for feature extraction and to build a scoring function based on words found in positive and negative reviews. In (Tan et al., 2008; Qiu et al., 2009), a dictionary containing commonly used words in expressing sentiment is employed to label a portion of informative examples from a given domain in order to reduce the labelling effort and to use the labelled documents as training set for a supervised classifier. Further, lexical information about associations between words and classes can be exploited and refined for specific domains by means of training examples to enhance accuracy (Melville et al., 2009). Finally, term weighting could foster sentiment classification as well, just like it happens in other mining tasks, from the general information retrieval to specific contexts, such as prediction of gene function annotations in biology (Domeniconi et al., 2015a). For this purpose, some researchers propose different term weighting schemes: a variant of the well-known *tf-idf* (Paltoglou and Thelwall, 2010), a supervised scheme based on both the importance of a term in a document and the importance of a term in expressing sentiment (Deng et al., 2014), regularised entropy in combination with singular term cutting and bias term in order to reduce the over-weighting issue (Wu and Gu, 2014).

With reference to cross-domain setting, a bunch of methods have been attempted to address the transfer learning issue.

Following works are based on some kind of supervision. In (Aue and Gamon, 2005), some approaches are tried in order to customise a classifier to a new target domain: training on a mixture of labelled data from other domains where such data are available, possibly considering just the features observed in target domain; using multiple classifiers trained on labelled data from diverse domains; including a small amount of labelled data from target. (Bollegala et al., 2013) suggest the adoption of a thesaurus containing labelled data from source domain and unlabelled data from both source and target domains. (Blitzer et al., 2007) discover a measure of domain similarity contributing to a better domain adaptation. (Pan et al., 2010) advance a spectral feature alignment algorithm which aims to align words belonging to different domains into same clusters, by means of domain-independent terms. These clusters form a latent space which can be used to improve sentiment classification accuracy of target domain. (He et al., 2011) extend the *joint sentiment-topic* model by adding prior words sentiment, thanks to the modification of the topic-word Dirichlet priors. Feature and document expansion are performed through adding polarity-bearing topics to align domains.

On the other hand, document sentiment classification may be performed by using unsupervised methods as well. In this case, most features are words commonly used in expressing sentiment. For instance, an algorithm is introduced to basically evaluate mutual information between the given sentence and two words taken as reference: *excellent* and *poor* (Turney, 2002). Furthermore, in another work not only a dictionary of words annotated with both their semantic polarity and their weights is built, but it also includes intensification and negation (Taboada et al., 2011).

Markov chain theory, whose a brief overview can be found in section 3, has been successfully applied in several text mining contexts, such as *information retrieval, sentiment analysis, text classification.*

Markov chains are particularly suitable for modelling hypertexts, which in turn can be seen as graphs, where pages or paragraphs represent states and links represent state transitions. This helps in some information retrieval tasks, because it allows discovering the possible presence of patterns when humans search for information in hypertexts (Qiu, 1993), performing link prediction and path analysis (Sarukkai, 2000) or, even, defining a ranking of Web pages just dealing with their hypertext structure, regardless information about page content (Page et al., 1999).

Markov chains, in particular hidden Markov chains, have been also employed to build information retrieval systems where firstly query, document or both are expanded and secondly the most relevant documents with respect to a given query are retrieved (Mittendorf and Schäuble, 1994; Miller et al., 1999), possibly in a *spoken document retrieval* context (Pan et al., 2012) or in the *cross-lingual area* (Xu and Weischedel, 2000). Anyhow, to fulfil these purposes, Markov chains are exploited to model term relationships. Specifically, they are used either in a single-stage or in a multi-stage fashion, the latter just in case indirect word relationships need to be modelled as well (Cao et al., 2007).

The idea of modelling word dependencies by means of Markov chains is also pursued for *sentiment analysis*. In practice, hidden Markov models (HMMs) aim to find out opinion words (i.e. words expressing sentiment) (Li et al., 2010), possibly trying to corre-

late them with particular topics (Mei et al., 2007; Jo and Oh, 2011). Typically, transition probabilities and output probabilities between states are estimated by using the Baum-Welch algorithm, whereas the most likely sequence of topics and related sentiments is computed through the Viterbi algorithm. The latter algorithm also helps in *Part-of-speech (POS) tagging*, where Markov chain states not only model terms but also tags (Jin et al., 2009; Nasukawa and Yi, 2003). In fact, when a tagging for a sequence of words is demanded, the goal is to find the most likely sequence of tags for that sequence of words.

Following works are focused on *text classification*, where the most widespread approach based on Markov models consists in building a HMM for each different category. The idea is, for each given document, to evaluate the probability of being generated by each HMM, finally assigning to that document the class corresponding to the HMM maximizing this probability (Yi and Beheshti, 2008; Xu et al., 2006; Vieira et al., ; Yi and Beheshti, 2013). Beyond directly using HMMs to perform text categorisation, they can also be exploited to model inter-cluster associations. For instance, words in documents can be clustered for dimensionality reduction purposes and each cluster can be mapped to a different Markov chain state (Li and Dong, 2013). Another interesting application is the classification of multi-page documents where, modelling each page as a different bag-of-words, a HMM can be exploited to mine correlation between documents to be classified (i.e. pages) by linking concepts in different pages (Frasconi et al., 2002).

3 METHOD DESCRIPTION

In this section, our novel method based on Markov chain for in-domain and cross-domain sentiment classification is introduced.

First of all, we would like to remind that a Markov chain is a mathematical model that is subject to transitions from one state to another in a states space $\mathcal{S}$. In particular, it is a stochastic process characterised by the so called *Markov property*, namely, future state only depends on current state, whereas it is independent from past states.

Before talking about our method in detail, notice that the entire algorithm can be split into three main stages, namely, the text pre-processing phase, the learning phase and the classification phase. We argue that the learning phase and the classification phase are the most innovative parts of the whole algorithm, because they accomplish both transfer learning and sentiment classification by means of only one abstraction, that is, the Markov chain.

3.1 Text Pre-processing Phase

The first stage of the algorithm is text pre-processing. Starting from a corpus of documents written in natural language, the goal is to transform them in a more manageable, structured format.

Initially, standard techniques are applied to the plain text, such as word tokenisation, punctuation removal, number removal, case folding, stopwords removal and the Porter stemming algorithm (Porter, 1980). Notice that stemming definitely helps the sentiment classification process, because words having the same morphological root are likely to be semantically similar.

The representation used for documents is the common bag-of-words, that is, a term-document matrix where each document d is seen as a multiset (i.e. bag) of words (or terms). Let $\mathcal{T} = \{t_1, t_2, ..., t_k\}$, where k is the cardinality of $\mathcal{T}$, be the dictionary of terms to be considered, which is typically composed of every term appearing in any document in the corpus to be analysed. In each document d, each word t is associated to a weight w_t^d, usually independent from its position inside d. More precisely, w_t^d only depends on *term frequency* $f(t,d)$, that is, the number of occurrences of t in document d, and in particular, represents *relative frequency* $rf(t,d)$, computed as follows:

$$w_t^d = rf(t,d) = \frac{f(t,d)}{\sum_{\tau \in \mathcal{T}} f(\tau,d)} \quad (1)$$

After having built the bags of words, a feature selection process is performed to limit the curse of dimensionality. On the one hand, feature selection allows selecting only the most profitable terms for the classification process. On the other hand, being k higher the more the dataset to be analysed is large, selecting only a small subset of the whole terms cuts down the computational burden required to perform both the learning phase and the classification phase.

Feature selection is an essential task, which could considerably affect the effectiveness of the whole classification method. Thus, it is critical to pick out a feature set as good as possible to support the classification process. In this paper, we try to tackle this issue by following some different feature selection approaches. Below, we briefly describe these methods, while the performed experiments where they are compared are presented in the next section.

3.1.1 Feature Selection Variants

The first feature selection method we use for testing is based on *document frequency*, $df(t)$, defined as:

$$df(t) = |\{d : t \in d\}| \quad (2)$$

Each term having a df higher than an established threshold is selected.

The second alternative consists in selecting only the terms included in a list (i.e. opinion wordlist) of commonly used words in expressing sentiment, regardless of their frequency, their document frequency or other aspects. These words simply are general terms that are known to have a certain polarity. The opinion wordlist used in this paper for English documents is that proposed in (Liu, 2012), containing 2003 positive words and 4780 negative words.

The previously presented feature selection methods are both unsupervised. On the contrary, we present straight away another viable option to perform the same task exploiting the knowledge of class labels. First of all, we use a supervised scoring function to find the most relevant features with respect to their ability to characterize a certain category. This function is either *information gain IG* or *chi-square* χ^2, defined as in (Domeniconi et al., 2015c). The ranking obtained as output is used on the one hand to select the best n features and on the other hand to change term weighting inside documents. In fact, this score $s(t)$ is a global value, stating the relevance of a certain word, whereas relative frequency, introduced by equation 1, is a local value only measuring the relevance of a word inside a particular document. Therefore, these values can be combined into a novel, different term weighting to be used for the bag-of-words representation, so that the weight w_t^d comes to be

$$w_t^d = rf(t,d) \cdot s(t) \quad (3)$$

Thus, according to the equation 3, both factors (i.e. the global relevance and the local relevance) may be taken into account.

3.2 Learning Phase

The learning phase is the second stage of our algorithm. As in any categorisation problem, the primary goal is to learn a model from a training set, so that a test set can be accordingly classified. Though, the mechanism should also allow transfer learning in case of cross-domain setting.

The basic idea consists in modelling term co-occurrences: the more words co-occur in documents the more their connection should be stronger. We could represent this scenario as a graph whose nodes represent words and whose edges represent the strength of the connections between them. Considering a document corpus $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ and a dictionary $\mathcal{T} = \{t_1, t_2, ..., t_k\}$, $A = \{a_{ij}\}$ is the set of connection weights between the term t_i and the term t_j and each a_{ij} can be computed as follows:

$$a_{ij} = a_{ji} = \sum_{d=1}^{N} w_{t_i}^d \cdot w_{t_j}^d \quad (4)$$

The same strategy could be followed to find the polarity of a certain word, unless having an external knowledge base which states that a word is intrinsically positive, negative or neutral. Co-occurrences between words and classes are modelled for each document whose polarity is given. Again, a graph whose nodes are either terms or classes and whose edges represent the strength of the connections between them is suitable to represent this relationship. In particular, given that $\mathcal{C} = \{c_1, c_2, ..., c_M\}$ is the set of categories and $B = \{b_{ij}\}$ is the set of edges between a term t_i and a class c_j, the strength of the relationship between a term t_i and a class c_j is augmented if t_i occurs in documents belonging to the set $D^j = \{d \in \mathcal{D} : c_d = c_j\}$.

$$b_{ij} = \sum_{d \in D^j} w_{t_i}^d \quad (5)$$

Careful readers may have noticed that the graph representing both term co-occurrences and term-class co-occurrences can be easily interpreted as a Markov chain. In fact, graph vertices are simply mapped to Markov chain nodes and graph edges are split into two directed edges (i.e. the edge linking states t_i and t_j is split into one directed edge from t_i to t_j and another directed edge from t_j to t_i). Moreover, for each state a normalisation step of all outgoing arcs is enough to satisfy the probability unitarity property. Finally, Markov property surely holds because each state only depends on directly linked states, since we evaluate co-occurrences considering just two terms (or a term and a class) at a time.

Now that we have introduced the basic idea behind our method, we explain how the learning phase is performed. We rely on the assumption that there exist a subset of common terms between source and target domain that can act as a bridge between domain specific terms, allowing and supporting transfer learning. So, these common terms are the key to let information about classes flow from source specific terms to target specific terms, exploiting term co-occurrences, as shown in Figure 1.

We would like to point out that the just described transfer learning process is not an additional step to be added in cross-domain problems; on the contrary, it is implicit in the Markov chain mechanism and, as such,

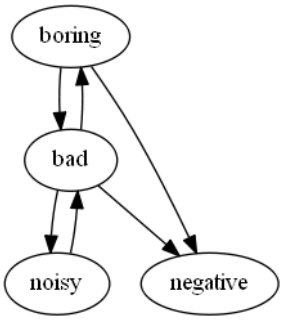

Figure 1: Transfer learning from a book specific term like *boring* to an electrical appliance specific term like *noisy* through a common term like *bad*.

it is performed in in-domain problems as well. Obviously, if both training set and test set are extracted from the same domain, it is likely that most of the terms in test set documents already have a polarity.

Apart from transfer learning, the Markov chain we propose also fulfils the primary goal of the learning phase, that is, to build a model that can be subsequently used in the classification phase. Markov chain can be represented as a transition matrix (MCTM), composed of four logically distinct submatrices, as shown in Table 1. It is a $(k+M)\times(k+M)$ matrix, having current states as rows and future states as columns. Each entry represents a transition probability, which is computed differently depending on the type of current and future states (term or class), as described below.

Table 1: This table shows the structure of MCTM. It is composed of four submatrices, representing transition probability that, starting from a current state (i.e. row), a future state (i.e. column) is reached. Both current states and future states can be either terms or classes.

	$\mathbf{t_1},...,\mathbf{t_k}$	$\mathbf{c_1},...,\mathbf{c_M}$
$\mathbf{t_1},...,\mathbf{t_k}$	$A^{'}$	$B^{'}$
$\mathbf{c_1},...,\mathbf{c_M}$	E	F

Let $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ be the subsets of document corpus $\mathcal{D}$ chosen as training set and test set respectively. The set A, whose each entry is defined by equation 4, is rewritten as

$$a_{ij} = a_{ji} = \begin{cases} 0, & i = j \\ \sum\limits_{d\in\mathcal{D}_{train}\cup\mathcal{D}_{test}} w^d_{t_i}\cdot w^d_{t_j}, & i \neq j \end{cases} \quad (6)$$

and the set B, whose each entry is defined by equation 5, is rewritten as

$$b_{ij} = \sum_{d\in D^j_{train}} w^d_{t_i} \quad (7)$$

where $D^j_{train} = \{d \in \mathcal{D}_{train} : c_d = c_j\}$. The submatrices $A^{'}$ and $B^{'}$ are the normalised forms of equations 6 and 7, computed so that each row of the Markov chain satisfies the probability unitarity property. Instead, each entry of the submatrices E and F looks like as follows:

$$e_{ij} = 0 \quad (8)$$

$$f_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (9)$$

Notice that E and F deal with the assumption that classes are absorbing states, which can never be left once reached.

3.3 Classification Phase

The last step of the algorithm is the classification phase. The aim is classifying test set documents by using the model learnt in the previous step. According to the bag-of-words representation, a document $d_t \in \mathcal{D}_{test}$ to be classified can be expressed as follows:

$$d_t = (w^{d_t}_{t_1},...,w^{d_t}_{t_k},c_1,...,c_M) \quad (10)$$

$w^{d_t}_{t_1},...,w^{d_t}_{t_k}$ is the probability distribution representing the initial state of the Markov chain transition matrix, whereas $c_1,...,c_M$ are trivially set to 0. We initially hypothesize to be in many different states (i.e. every state t_i so that $w^{d_t}_{t_i} > 0$) at the same time. Then, simulating a single step inside the Markov chain transition matrix, we obtain a posterior probability distribution not only over terms, but also over classes. In such a way, estimating the posterior probability that d_t belongs to a certain class c_i, we could assign to d_t the most likely label $c_i \in \mathcal{C}$. The posterior probability distribution after one step in the transition matrix, starting from document d_t, is:

$$d^*_t = (w^{d^*_t}_{t_1},...,w^{d^*_t}_{t_k},c^*_1,...,c^*_M) = d_t \times MCTM \quad (11)$$

where d_t is a column vector having size $(k+M)$ and $MCTM$ is the Markov chain transition matrix, whose size is $(k+M)\times(k+M)$. At this point, the category that will be assigned to d_t is computed as follows:

$$c_{d_t} = \arg\max_{i\in C^*} c^*_i \quad (12)$$

where $C^* = \{c^*_1,...,c^*_M\}$ is the posterior probability distribution over classes.

3.4 Computational Complexity

The computational complexity of our method is the time required to perform both the learning phase and the classification phase. Regarding the learning phase, the computational complexity overlaps with

the time needed to build the Markov chain transition matrix, say $time(MCTM)$, which is

$$time(MCTM) = time(A) + time(B) + \\ + time(A^{'} + B^{'}) + time(E) + time(F) \quad (13)$$

Remember that A and B are the submatrices representing the state transitions having a term as current state. Similarly, E and F are the submatrices representing the state transitions having a class as current state. $time(A^{'} + B^{'})$ is the temporal length of the normalisation step, necessary in order to observe the probability unitarity property. On the other hand, E and F are simply a null and an identity matrix, requiring no computation. Thus, since time complexity depends on these factors, all should be estimated.

The only assumption we can do is that in general $|\mathcal{T}| >> |\mathcal{C}|$. The time needed to compute A is $O(\frac{|\mathcal{T}|^2}{2} \cdot (|\mathcal{D}_{train}| + |\mathcal{D}_{test}|))$, which in turn is equal to $O(|\mathcal{T}|^2 \cdot (|\mathcal{D}_{train}| + |\mathcal{D}_{test}|))$. Regarding transitions from terms to classes, building the submatrix B requires $O(|\mathcal{T}| \cdot |\mathcal{C}| \cdot |\mathcal{D}_{train}|)$ time. In sentiment classification problems we could also assume that $|\mathcal{D}| >> |\mathcal{C}|$ and, as a consequence, the previous time becomes $O(|\mathcal{T}| \cdot |\mathcal{D}_{train}|)$. The normalisation step, which has to be computed one time only for both A and B, is $O(|\mathcal{T}| \cdot (|\mathcal{T}| + |\mathcal{C}|) + |\mathcal{T}| + |\mathcal{C}|) = O((|\mathcal{T}| + 1) \cdot (|\mathcal{T}| + |\mathcal{C}|))$, which can be written as $O(|\mathcal{T}|^2)$ given that $|\mathcal{T}| >> |\mathcal{C}|$. Further, building the submatrix E requires $O(|\mathcal{T}|^2)$ time, whereas for submatrix F $O(|\mathcal{T}| \cdot |\mathcal{C}|)$ time is needed, which again can be written as $O(|\mathcal{T}|)$ given that $|\mathcal{T}| >> |\mathcal{C}|$. Therefore, the overall complexity of the learning phase is

$$time(MCTM) \simeq time(A) = \\ = O(|\mathcal{T}|^2 \cdot (|\mathcal{D}_{train}| + |\mathcal{D}_{test}|)) \quad (14)$$

In the classification phase, two operations are performed for each document to be categorised: the matrix product in equation 11, which requires $time(MatProd)$, and the maximum computation in equation 12, which requires $time(Max)$. Hence, as we can see below

$$time(CLASS) = time(MatProd) + time(Max) \quad (15)$$

the classification phase requires a time that depends on the previous mentioned factors. The matrix product can be computed in $O((|\mathcal{T}| + |\mathcal{C}|)^2 \cdot |\mathcal{D}_{test}|)$ time, which can be written as $O(|\mathcal{T}|^2 \cdot |\mathcal{D}_{test}|)$ given that $|\mathcal{T}| >> |\mathcal{C}|$. On the other hand, the maximum requires $O(|\mathcal{C}| \cdot |\mathcal{D}_{test}|)$ time. Since the assumption that $|\mathcal{T}| >> |\mathcal{C}|$ still holds, the complexity of the classification phase can be approximated by the calculus of the matrix product.

Lastly, the overall complexity of our algorithm, say $time(Algorithm)$, is as follows:

$$time(Algorithm) = time(MCTM) + time(CLASS) \simeq \\ \simeq time(MCTM) = O(|\mathcal{T}|^2 \cdot (|\mathcal{D}_{train}| + |\mathcal{D}_{test}|)) \quad (16)$$

This complexity is comparable to those we have estimated for the other methods, which are compared in the upcoming experiments section.

4 EXPERIMENTS

The Markov chain based method has been implemented in a framework entirely written in Java. Algorithm performance has been evaluated through the comparison with *Spectral feature alignment* (*SFA*) by Pan et al. (Pan et al., 2010) and *Joint sentiment-topic model with polarity-bearing topics* (*PBT*) by He et al. (He et al., 2011), which, to the best of our knowledge, currently are the two best performing approaches.

We used a common benchmark dataset to be able to compare results, namely, a collection of Amazon[1] reviews about four domains: Book (B), DVD (D), Electronics (E) and Kitchen appliances (K). Each domain contains 1000 positive and 1000 negative reviews written in English. The text pre-processing phase described in 3.1 is applied to convert plain text into the bag-of-words representation. Then, before the learning phase and the classification phase introduced in 3.2 and 3.3 respectively, we perform feature selection in accordance with one of the alternatives presented in 3.1.1. The goodness of results is measured by accuracy, averaged over 10 different source-target splits.

Performances with every feature selection method are shown below and compared with the state of the art. Differently, the Kitchen domain is excluded from both graphics and tables due to space reasons. Anyway, accuracy values are aligned with the others, so that considerations to be done do not change.

4.1 Setup and Results

The approaches we compare only differ in the applied feature selection method: document frequency, opinion wordlist and supervised feature selection technique with term ranking in the Markov chain.

In the first experiment we perform, as shown in Figure 2, the only parameter to be set is the minimum document frequency df that a term must have in order to be selected, varied from 25 to 100 with step length

[1] www.amazon.com

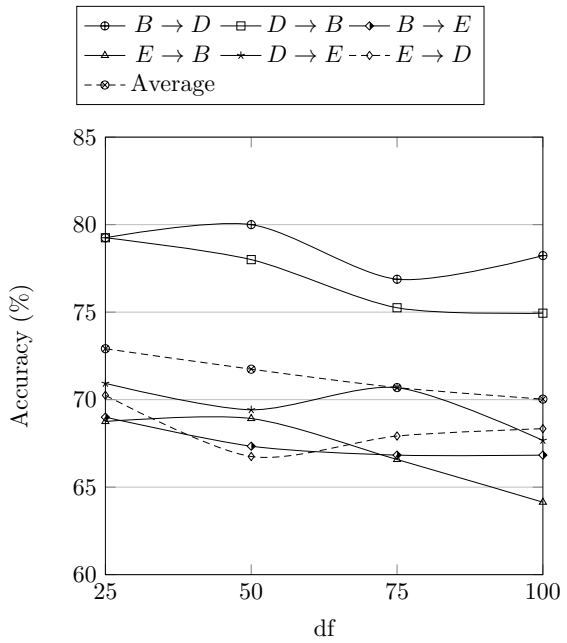

Figure 2: Cross-domain classification by varying the minimum df to be used for feature selection.

25. In other words, setting the minimum df equal to n means that terms occurring in less than n documents are ruled out from the analysis. Each line in Figure 2 represents the accuracy trend for a particular source-target couple, namely, $B \rightarrow D$, $D \rightarrow B$, $B \rightarrow E$, $E \rightarrow B$, $D \rightarrow E$, $E \rightarrow D$; further, an extra line is visible, portraying the average trend.

We can see that accuracy decreases on average when minimum document frequency increases. This probably is due to the fact that document frequency is an unsupervised technique; therefore, considering smaller feature sets (i.e. higher df values) comes out not to be effective in classifying target documents. Unsupervised methods do not guarantee that the selected features help in discriminating among categories and, as such, a bigger dictionary could support the learning phase. Further, accuracy is lower anytime E is considered, because E is a completely different domain with respect to B and D, which instead have more common words. Therefore, it is increasingly important to select the best possible feature set in order to help transfer learning.

An alternative to unsupervised methods consists in choosing the Bing Liu wordlist as dictionary. In this case, no parameter needs to be tuned but Porter stemmer has to be applied to the wordlist so that terms inside documents match those in the wordlist. Comparing the performance of our algorithm when using opinion words as features with that achieved using terms having minimum $df = 25$ (Figure 3), we may

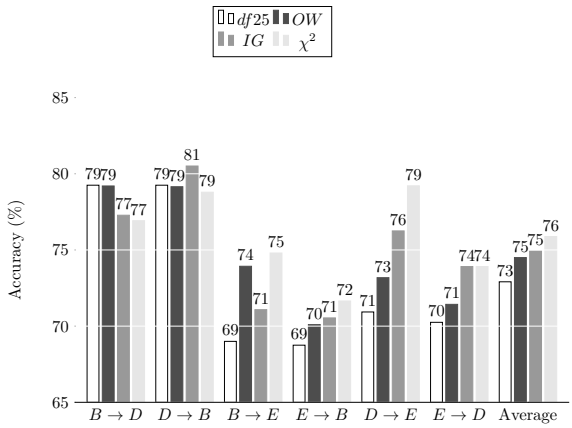

Figure 3: Cross-domain classification by comparing some feature selection methods, such as the unsupervised minimum document frequency df, the opinion wordlist and the supervised information gain IG and chi-square χ^2 scoring functions. Minimum $df = 25$ is set regarding the unsupervised method. Instead, the best 250 features are selected in accordance with the supervised scoring functions.

notice that the former outperforms the latter on average. This outcome suggests that, when features have a stronger polarity, inferred by the source domain, transfer learning is eased and our algorithm achieves better performance.

The Bing Liu wordlist only contains opinion words, which are general terms having well-known polarity. Nevertheless, there may be other words, possibly domain dependent, fostering the classification process. For this purpose, supervised feature selection techniques can be exploited to build the dictionary to be used. We would like to remark that, including domain dependent terms, the transfer learning capability of our method is increasingly important to let information about polarity flow from source domain to target domain terms.

Below, two tests are presented with reference to supervised feature selection techniques: in the former (Figure 4), features are selected by means of chi-squared (χ^2), varying the number of selected features from 250 to 1000 with step length 250; in the latter (Figure 3), the two supervised feature selection techniques mentioned in 3.1.1 are compared. Figure 4 reveals that there are no relevant variations on average in performance by increasing the number of features to be selected. On the one hand, this means that 250 words are enough to effectively represent the source-target couple, reminding that these are the best features according to the supervised method used. On the other hand, this proves that our algorithm produces stable results by varying the number of features to be selected.

Figure 3 shows that on average the usage of a supervised feature selection technique is better than se-

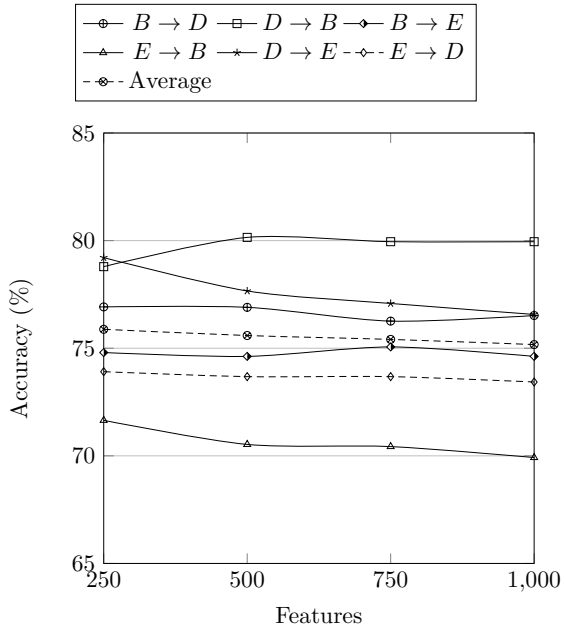

Figure 4: Cross-domain classification by varying the number of features to be selected in a supervised way. χ^2 scoring function is used in order to select features.

lecting just well-known opinion words. This confirms the ability of our method in performing transfer learning. Further, the same configuration, where 250 features are selected by using χ^2 scoring function, also achieves better accuracy than using *IG* to perform the feature selection process (Figure 3). Summing up, our Markov chain based method achieves its best performance in Amazon datasets by selecting the best 250 features by means of χ^2 scoring function in the text pre-processing phase.

Table 2 displays a comparison between our new method, referred as *MC*, and other works, namely *SFA* and *PBT*. Accuracy values are comparable, despite both *SFA* and *PBT* perform better on average. On the other hand, we would like to put in evidence that our algorithm only requires 250 features, whereas *PBT* needs 2000 features and *SFA* more than 470000. Therefore, growing the computational complexity of the three approaches quadratically with the number of features, the convergence of our algorithm is supposed to be faster even if this hypothesis cannot be proved without implementing the other methods.

Finally, in Table 3 we can see that similar considerations can be done in an in-domain setting. Notice that nothing needs to be changed in our method to perform in-domain sentiment classification, whereas other works use standard classifiers completely bypassing the transfer learning phase.

Table 2: Results in cross-domain sentiment classification, compared with other works. For each dataset, the best accuracy is in bold.

	MC	SFA	PBT
$B \to D$	76.92%	**81.50%**	81.00%
$D \to B$	78.79%	78.00%	**79.00%**
$B \to E$	74.80%	72.50%	**78.00%**
$E \to B$	71.65%	**75.00%**	73.50%
$D \to E$	**79.21%**	77.00%	79.00%
$E \to D$	73.91%	**77.50%**	76.00%
Average	75.88%	76.92%	77.75%

Table 3: Results in in-domain sentiment classification, compared with other works. For each dataset, the best accuracy is in bold.

	MC	SFA	PBT
$B \to B$	76.77%	**81.40%**	79.96%
$D \to D$	**83.50%**	82.55%	81.32%
$E \to E$	80.90%	**84.60%**	83.61%
Average	80.39%	82.85%	81.63%

5 CONCLUSIONS

We introduced a novel method for in-domain and cross-domain sentiment classification relying on Markov chain theory. We proved that this new technique not only fulfils the categorisation task, but also allows transfer learning from source domain to target domain in cross-domain setting. The algorithm aims to build a Markov chain transition matrix, where states represent either terms or classes, whose co-occurrences in documents, in turn, are employed to compute state transitions. Then, a single step in the matrix is performed for the sake of classifying a test document.

We compared our approach with other two works and we showed that it achieves comparable performance in term of effectiveness. On the other hand, lower parameter tuning is required than previous works, since only the pre-processing parameters need to be calibrated. Finally, in spite of having a comparable computational complexity, growing quadratically with the number of features, much fewer terms are demanded to obtain good accuracy.

Possible future works on the one side could aim to extend its applicability to other text mining tasks, such as for example text categorisation, and on the other side could face the problem of improving the algorithm effectiveness. In fact, our method is absolutely general and could be applied as is to text categorisation. Moreover, so far we have assessed performance only in 2-classes sentiment classification; therefore, a 3-classes setting (i.e. adding the neutral

category) could be tested. Further, since our algorithm only relies on term co-occurrences, its applicability can be easily extended to other languages. Instead, to improve the algorithm effectiveness, transfer learning needs to be strengthened: for this purpose, a greater number of steps in the Markov chain transition matrix during the classification phase could help.

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)*, volume 1, pages 2–1.

Blitzer, J., Dredze, M., Pereira, F., et al. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447.

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

Cao, G., Nie, J.-Y., and Bai, J. (2007). Using markov chains to exploit word relationships in information retrieval. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, pages 388–402. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Dai, W., Xue, G.-R., Yang, Q., and Yu, Y. (2007). Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 210–219. ACM.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

Deng, Z.-H., Luo, K.-H., and Yu, H.-L. (2014). A study of supervised term weighting scheme for sentiment analysis. *Expert Systems with Applications*, 41(7):3506–3513.

Domeniconi, G., Masseroli, M., Moro, G., and Pinoli, P. (2015a). Random perturbations and term weighting of gene ontology annotations for unknown gene function discovering. In *Fred, A. et al. (eds.) IC3K 2014. CCIS*, volume 553. Springer.

Domeniconi, G., Moro, G., Pasolini, R., and Sartori, C. (2014). Cross-domain text classification through iterative refining of target categories representations. In *Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval*.

Domeniconi, G., Moro, G., Pasolini, R., and Sartori, C. (2015b). Iterative refining of category profiles for nearest centroid cross-domain text classification. In *Fred, A. et al. (eds.) IC3K 2014. CCIS*, volume 553. Springer.

Domeniconi, G., Moro, G., Pasolini, R., and Sartori, C. (2015c). A study on term weighting for text categorization: a novel supervised variant of tf.idf. In *Proceedings of the 4th International Conference on Data Management Technologies and Applications*.

Frasconi, P., Soda, G., and Vullo, A. (2002). Hidden markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems*, 18(2-3):195–217.

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.

Jin, W., Ho, H. H., and Srihari, R. K. (2009). Opinionminer: a novel machine learning system for web opinion mining and extraction. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1195–1204. ACM.

Jo, Y. and Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM.

Li, F. and Dong, T. (2013). Text categorization based on semantic cluster-hidden markov models. In *Advances in Swarm Intelligence*, pages 200–207. Springer.

Li, F., Huang, M., and Zhu, X. (2010). Sentiment analysis with global topics and local dependency. In *AAAI*, volume 10, pages 1371–1376.

Li, L., Jin, X., and Long, M. (2012). Topic correlation analysis for cross-domain text classification. In *AAAI*.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM.

Melville, P., Gryc, W., and Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284. ACM.

Miller, D. R., Leek, T., and Schwartz, R. M. (1999). A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221. ACM.

Mittendorf, E. and Schäuble, P. (1994). Document and passage retrieval based on hidden markov models. In *SIGIR94*, pages 318–327. Springer.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.

Paltoglou, G. and Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395. Association for Computational Linguistics.

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*, pages 751–760. ACM.

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

Pan, Y.-C., Lee, H.-Y., and Lee, L.-S. (2012). Interactive spoken document retrieval with suggested key terms ranked by a markov decision process. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(2):632–645.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Qiu, L. (1993). Markov models of search state patterns in a hypertext information retrieval system. *Journal of the American Society for Information Science*, 44(7):413–427.

Qiu, L., Zhang, W., Hu, C., and Zhao, K. (2009). Selc: a self-supervised model for sentiment classification. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 929–936. ACM.

Sarukkai, R. R. (2000). Link prediction and path analysis using markov chains. *Computer Networks*, 33(1):377–386.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Tan, S., Wang, Y., and Cheng, X. (2008). Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 743–744. ACM.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Vieira, A. S., Iglesias, E. L., and Diz, L. B. Study and application of hidden markov models in scientific text classification.

Wu, H. and Gu, X. (2014). Reducing over-weighting in supervised term weighting for sentiment analysis. COLING.

Xu, J. and Weischedel, R. (2000). Cross-lingual information retrieval using hidden markov models. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 95–103. Association for Computational Linguistics.

Xu, R., Supekar, K., Huang, Y., Das, A., and Garber, A. (2006). Combining text classification and hidden markov modeling techniques for structuring randomized clinical trial abstracts. In *AMIA Annual Symposium Proceedings*, volume 2006, page 824. American Medical Informatics Association.

Xue, G.-R., Dai, W., Yang, Q., and Yu, Y. (2008). Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 627–634. ACM.

Yi, K. and Beheshti, J. (2008). A hidden markov model-based text classification of medical documents. *Journal of Information Science*.

Yi, K. and Beheshti, J. (2013). A text categorization model based on hidden markov models. In *Proceedings of the Annual Conference of CAIS/Actes du congrès annuel de l'ACSI*.