SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 27 (2011) 248 – 257

Procedia Social and Behavioral Sciences

Pacific Association for Computational Linguistics (PACLING 2011)

Merged Agreement Algorithms for Domain Independent Sentiment Analysis

Dinko Lambov[a*], Sebastião Pais[a,c], Gãel Dias[a,b]

[a]HULTIG - University of Beira Interior, Portugal
[b]DLU/GREYC - University of Caen Basse-Normandie, France
[c]CRI-Ecole Nationale Superieure des Mines de Paris, France

Abstract

In this paper, we consider the problem of building models that have high sentiment classification accuracy across domains. For that purpose, we present and evaluate three new algorithms based on multi-view learning using both high-level and low-level views, which show improved results compared to the state-of-the-art SAR algorithm [1] over cross-domain text subjectivity classification. Our experimental results present accuracy levels of 80% with two views, combining SVM classifiers over high-level features and unigrams compared to 77.1% for the SAR algorithm.

Keywords: Sentiment Analysis; Subjectivity Classification; Multi-view Learning

1. Introduction

Over the past few years, there has been an increasing number of publications focused on the classification of sentiment in texts. However, as stated in [2, 3, 4, 5], most research have focused on the construction of models within particular domains and have shown difficulties to cross thematic spheres. Within this context, three main approaches have been tackled. The first one is to train a classifier on a domain-mixed set of data as in [2, 5]. The second solution is to propose high-level features, which do not depend so much on topics such as part-of-speech statistics or other semantic resources as in [3, 6]. In this case, high level representations do not reflect the topic of the document, but rather the text genre. The third solution is to propose multi-view learning algorithms. The basic idea is to train at least two classifiers on one source domain and then update the set of labelled examples with new examples from a target domain when both classifiers agree on the class of the unlabeled example. This process is then iterated until convergence.

* Corresponding author:
Email address: dinko@hultig.di.ubi.pt (Dinko Lambov), sebastiao@hultig.di.ubi.pt (Sebastiao˜ Pais), ddg@hultig.di.ubi.pt (Gael¨ Dias)

doi:10.1016/j.sbspro.2011.10.605

In parallel, [7] proposes a multi-view learning approach to improve the classification accuracy of polarity identification of Chinese product reviews based on translated English reviews. For that purpose, they use the co-training algorithm [8] with an agreement constraint combined with two SVM classifiers. In this paper, we propose to compare the SAR algorithm [1] with the co-training algorithm strategy constrained by agreement (i.e. a multi-view learning paradigm) as in [7]. Within this context, we propose three multi-view learning algorithms, which best one presents accuracy levels across domains of 80% compared to 77.1% for the SAR.

2. Characterizing Subjectivity

Our methodology aims at classifying texts at the subjectivity level (i.e. subjective vs. objective) taking into account both high-level features (e.g. level of abstraction of nouns or level of subjective adjectives), which easily cross domains as shown in [6] as well as low-level features (e.g. unigrams or bigrams), which evidence high precision results within domains [9].

2.1. High-Level Features

High-level features are usually used to cross domain as they do not depend so much on topics and usually define text genre. In this paper, we propose seven high-level features, which are described below.

Intensity of Affective Words: sentiment expressions mainly depend on some words, which can express subjective sentiment orientation. Within this context, [10] use words from the WordNet Affect Lexicon [11] to annotate emotions. So, we propose to evaluate the level of affective words in texts as shown in Equation 1 by using the WordNet Affect Lexicon. Some examples of affective words are given in Table 1.

Table 1. Examples for Affective Words.

Affective Category	Words
Fury	furious, maddened, enraged, angered
Distress	worrying, disturbed, upset, worried, unhappy
Stupefaction	stupid, dazed, stun, baffle, amaze
Disgust	disgust, revolt, repel, sicken, wicked

$$K_1 = \frac{\text{total of affective words in text}}{\text{total of words in text}}. \quad (1)$$

Dynamic and Semantically Oriented Adjectives: [12] consider two features for the identification of opin-ionated sentences: (1) semantic orientation, which represents an evaluative characterization of word devi-ation from its semantic group and (2) dynamic adjectives, which characterize words' ability to express a property in varying degrees. In particular, semantic-oriented adjectives are polar words that are either pos-itive or negative and dynamic adjectives are adjectives with the "qualities that are thought to be subject to control by the possessor and hence can be restricted temporally". Some examples are given in Table 2. For the present study, we use the set of dynamic adjectives manually identified by [12] and the set of semantic orientation labels assigned as in [13]. So, we propose to evaluate the level of these adjectives in texts as shown in Equations 2 and 3.

Table 2. Examples for Dynamic and Semantically Oriented Adjectives.

Affective Category	Words
Dynamic Adjectives	abusive, careful, clever, foolish, brave
Semantically Oriented Adjectives	abnormal, banal, attractive, boring

$$K_2 = \frac{\text{total of dynamic adjectives in text}}{\text{total of adjectives in text}}. \quad (2)$$

$$K_3 = \frac{\text{total of semantic adjectives in text}}{\text{total of adjectives in text}}. \quad (3)$$

Classes of Verbs: [14] present a method using verb class information. According to her, verb classes express objectivity and polarity. To obtain relevant verb classes, they use InfoXtract [15], which groups verbs according to classes that correspond to their polarity. As InfoXtract is not freely available, we reproduced their methodology by using the classification of verbs available in Levin's English Verb Classes and Alter-nations [16]. Some examples are given in Table 3. So, we propose to evaluate the level of the following three classes of verbs: Conjecture, Marvel and See as in Equations 4, 5 and 6.

Table 3. Verb Examples for Levin's Verb Classes.

Class Verb	Verbs
Conjecture	admit, allow, deny, guess, show, suspect, assert, guarantee
See	detect, see, feel, smell, taste, sense, notice, hear, discern
Marvel	anger, fear, cry, care, bleed, bother, glory, heart, obsess, suffer

$$K_4 = \frac{\text{total of conjecture verbs in text}}{\text{total of verbs in text}}. \quad (4)$$

$$K_5 = \frac{\text{total of marvel verbs in text}}{\text{total of verbs in text}}. \quad (5)$$

$$K_6 = \frac{\text{total of see verbs in text}}{\text{total of verbs in text}}. \quad (6)$$

Level of Abstraction of Nouns: There is linguistic evidence that level of generality is a characteristic of opinionated texts, i.e. subjectivity is usually expressed in more abstract terms than objectivity [6]. Indeed, descriptive texts tend to be more precise and more objective and as a consequence more specific. In other words, a word is abstract when it has few distinctive features and few attributes that can be pictured in the mind. One way of measuring the abstractness of a word is by the hypernym relation in WordNet [17]. In particular, a hypernym metric can be the number of levels in a conceptual taxonomic hierarchy above a word (i.e. superordinate to). For example, chair (as a seat) has 7 hypernym levels chair) f urniture) f urnishings) instrumentality) arti f act) ob ject) entity in WordNet. So, a word having more hypernym levels is more concrete than one with fewer levels. So, we propose to evaluate the hypernym levels of all the nouns in texts as shown in Equation 7.

$$K_7 = \frac{\text{total of hypernym levels in text}}{\text{total of nouns in text}}. \quad (7)$$

Calculating the level of abstraction of nouns should be preceded by word sense disambiguation. Indeed, it is important that the correct sense is taken for the calculation of the hypernym level in WordNet. However, in practice, taking the most common sense of each word gives similar results as taking all the senses on average as shown in [6].

2.2. Low-Level Features

The most common set of features used for text classification is information regarding the occurrences of words or word ngrams in texts. Most of text classification systems treat documents as simple bags-of-words and use word counts as features. Here, we consider texts as bags of lemmatized unigrams or bigrams, for which we compute their TF.IDF weights as in Equation 8 where $w_{i\,j}$ is the weight of term j in document i, t $f_{i\,j}$ is the normalized frequency of term j in document i, N is the total number of documents in the collection, and n_j is number of documents where the term j occurs at least once.

$$w_{ij} = tf_{ij} * \log_2 \frac{N}{n_j}. \quad (8)$$

3.Subjective/ObjectiveText Datasets

To perform our experiments, we used three manually annotated standard corpora and built one corpus based on Web resources, which could be automatically annotated as objective or subjective.

3.1. Existing Resources for Sentiment Classification

The first resource is based on the Multi-Perspective Question Answering (MPQA) Opinion Corpus[1]. Based on the work done by [9] who propose to classify texts based only on their subjective/objective parts, we built a corpus of 100 objective (resp. subjective) texts by randomly selecting sentences containing only subjective or objective phrases. This case represents the "ideal" case where all the sentences in texts are either subjective or objective. The second corpus (RIMDB) is the subjectivity dataset v1.0[2], which contains 5000 subjective and 5000 objective sentences collected from movie reviews data [9]. Similarly to the MPQA corpus, we built a corpus of 100 objective (resp. subjective) texts by randomly selecting only subjective or objective sentences. The third corpus (CHES) was developed by [14] who manually annotated a data set of objective and subjective documents[3].

3.2. Automatic Construction of Labelled Dataset

For our dataset (WBLOG), we downloaded part of the static Wikipedia dump archive[4] and automatically spidered Weblogs from different domains. In fact, we propose to compare Wikipedia texts and Weblogs to reference objective and subjective corpora and show that Wikipedia texts are representative of objectivity and Weblogs are representative of subjectivity. For that purpose, we proposed an exhaustive evaluation based on (1) the Rocchio classification method [18] for different part-of-speech tag levels and (2) language modeling. In Table 4, we present the results of the Rocchio classification where the test vector is the set of Wikipedia sentences and the trained vectors are the subjective and objective sentences from the subjectivity v1.0 corpus (RIMDB). The results confirm our initial assumption that texts from Wikipedia convey objective contents, although the role of verbs seems less clear with respect to subjectivity as opposed to what is exposed in [14].

Table 4. Results with the Wikipedia Test Dataset.

Part-of-Speech	Subjective (RIMDB)	Objective (RIMDB)	Class
All Words	0.76	0.79	Objective
All ADJ	0.54	0.61	Objective
All V	0.71	0.67	Subjective
All N	0.66	0.69	Objective
All ADJ + All V	0.65	0.66	Objective
All ADJ + All N	0.65	0.68	Objective
All N + All V	0.70	0.69	Subjective
All ADJ + All N + All V	0.68	0.69	Objective

Similarly, we performed the same experiment where the test vector is the set of Weblogs sentences and the trained vectors are the subjective and objective sentences from the subjectivity v1.0 corpus (RIMDB). The results are presented in Table 5 and clearly show that at any part-of-speech level, Weblogs embody subjectivity.

[1]http://www.cs.pitt.edu/mpqa/
[2]http://www.cs.cornell.edu/People/pabo/movie-review-data/
[3]http://www.tc.umn.edu/ ches0045/data/
[4]http://download.wikimedia.org/enwiki/

Table 5. Results with the Weblogs Test Dataset.

Part-of-Speech	Subjective (RIMDB)	Objective (RIMDB)	Class
All Words	0.60	0.56	Subjective
All ADJ	0.52	0.49	Subjective
All V	0.53	0.48	Subjective
All N	0.47	0.43	Subjective
All ADJ + All V	0.49	0.48	Subjective
All ADJ + All N	0.48	0.44	Subjective
All N + All V	0.50	0.45	Subjective
All ADJ + All N + All V	0.47	0.46	Subjective

In spite of encouraging classifications, the values of the cosine similarity measure within the same morphological level between the trained and the test vectors are usually very close. This does not give much confidence in the results. For that purpose, we proposed another methodology based on language modeling. The basic idea is that objective and subjective languages are intrinsically different. Consequently, if we build a language model based on Weblogs, the subjective part of the subjectivity v1.0 corpus (RIMDB) should be more probable than the objective part, and vice and versa. This probability is transformed into perplexity (Px) and entropy (H) measures within the CMU-Toolkit[5]. The results of this experiment are given in Table 6 for a trigram language model.

Table 6. Results With Language Modelling.

	Wikipedia	Weblogs
Objective	$Px = 691.27$ - $H = 9.43$	$Px = 2027.06$ - $H = 10.99$
Subjective	$Px = 880.67$ - $H = 9.75$	$Px = 1991.09$ - $H = 10.96$

To summarize the results in Table 6, the trained model Wikipedia shows lower perplexity and entropy for the objective sentences than for the subjective sentences. The opposite happens when using the trained model Weblogs. In that case, lower perplexity and entropy are shown for the subjective sentences than for the objective sentences. Once again, our assumptions are confirmed as objective (resp. subjective) sentences are intrinsically closer to the Wikipedia (resp. Weblogs) model than subjective (resp. objective) ones. Indeed, the lower the perplexity and the entropy are, the closer to the model the sentences are. Thanks to this analysis, we are now able to automatically build large data sets of learning examples based on common sense judgments.

4. Multi-View Learning

While semi-supervised learning is usually associated to small labelled datasets situations and tries to automatically increase the number of labelled examples, multi-view learning aims at learning a compromise model of different views. A classical semi-supervised algorithm is the well-known co-training algorithm [8], which includes as new labelled examples the best classified examples by each classifier individually. [7] proposed a slight modification of the co-training algorithm by introducing an agreement constraint, which can be thought as a way of providing a multi-view learning approach. Within this context, new labelled examples are included in the set of labelled examples if all classifiers agree on their labels. As such, all classifiers tend to converge to a compromise learner. This is the definition of the multi-view paradigm. Different works have been proposed following this approach, but SAR [1] is certainly the best reference up-to-date for sentiment classification.

[5] *http://www.speech.cs.cmu.edu/SLM info.html*

4.1. The SAR Algorithm

[1] proposed the Stochastic Agreement Regularization (SAR) algorithm to deal with cross-domain po-larity classification. In particular, SAR models a probabilistic agreement framework based on minimizing the Bhattacharyya distance between models trained using two different views. It regularizes the models from each view by constraining the amount by which it allows them to disagree on unlabeled instances from a theoretical model. Their co-regularized objective, which has to be minimized, is defined in Equation 9 where L_i for i = 1::2 are the standard regularized log likelihood losses of the probabilistic models p_1 and p_2, $E_u[B(p_1(\ _1); p_2(\ _1))]$ is the expected Bhattacharyya distance between the predictions of the two models on the unlabeled data, and c is a constant defining the weight of the agreement between unlabeled data.

$$\text{Min } L_1(\theta_1) + L_2(\theta_2) + cE_u[B(p_1(\theta_1), p_2(\theta_1))]. \tag{9}$$

4.2. Merged Agreement Algorithms

As the SAR algorithm is based on agreement between different models, we propose three algorithms based on the well-known co-training by introducing different agreement constraints. On the one hand, we know that high-level features provide strong opinion evidence across domains [3, 6]. On the other hand, word-based models show remarkable results for in-domain classification tasks [9]. As a consequence, we expect that agreement between low-level classifiers and high-level classifiers will allow the classifiers to self-adapt to new domains.

4.2.1. The MAA and BMAA Algorithms

The Merged Agreement Algorithm (MAA) is an adaptation of the algorithm proposed in [7] and it is defined in Algorithm 1.

Algorithm 1 The MAA Algorithm.

```
Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain
Output: Trained classifier H2
H1.AgreeList ← {}
H2.AgreeList ← {}
for k iterations do
    Train a classifier H1 on view V1 of L
    Train a classifier H2 on view V2 of L
    Allow H1 and H2 to label U
    for all d ∈ U do
        if H1.Class[d] = H2.Class[d] then
            H1.AgreeList ← H1.AgreeList ∪ {< d; H1.Class[d] >}
            H2.AgreeList ← H2.AgreeList ∪ {< d; H2.Class[d] >}
        end if
    end for
    L ← L ∪ {the most confidently predicted P positive and N negative examples from H1 on U ∈
    H1.AgreeList}
    L ← L ∪ {the most confidently predicted P positive and N negative examples from H2 on U ∈
    H2.AgreeList}
end for
```

It is based on the co-training algorithm with agreement, but instead of just taking into account unlabeled examples with similar predictions from both classifiers to update the set of labelled examples such as in [7], we impose that only the examples with highest confidence upon agreement are added to the labelled list. Basically, the MAA takes two main inputs: a set of labelled examples from one domain (L), the source domain, and a set of unlabeled examples from another domain (U), the target domain. After training on the source domain, both classifiers classify unlabeled documents from the target domain. If both classifiers

agree on their predictions, the unlabeled document is added to an agree list for each classifier with the categorization label and the classification confidence. Finally, the P positive (subjective) and N negative (objective) documents with higher confidence values are selected from each agree list and transferred from the set of unlabeled documents to the labeled set. It is important to point at the fact that the MAA algorithm as the one proposed by [7] may produce unbalanced data sets. Indeed, from both agree lists of H1 and H2, we may update the labelled list with more positive examples than negative ones and vice and versa as classifiers may agree more on one class than another. Without a balancing parameter, it is not necessary to have a minimum number of positive or negative documents, which agree on labels for both classifiers. If H1 and H2 agree only on positive predictions then only positive examples will be added to L. As a consequence, we propose to modify the MAA to balance the parameter values of P and N at each iteration. So, if the number of predicted subjective or objective documents is equal to 0, it is used as a stopping criterion. Otherwise, the minimum number of positive or negative new labelled examples is chosen to update the source labelled example list L. This cycle is repeated for k iterations or until there are no positive or negative candidate documents in the agree lists. We call this method the Balanced Merged Agreement Algorithm (BMAA), our second algorithm proposal.

4.2.2. The BMAADR Algorithm

With the MAA and the BMAA algorithms, the most confidently predicted P and N examples from H1 and H2 are selected to update L. However, for example, we may update L with a positive example from H1, which agrees on the classification of H2 but where the difference between each confidence is high. As a consequence, we may update L with examples where only one of the classifiers is very confident about the classification although they agree on the classification. The idea of our new proposal is to measure an "average" confidence value for all examples for which there is agreement between classifiers so that the highest "on average" new labelled examples are added to L. For that purpose, after each classification on unlabeled data, both agree lists are sorted by decreasing classification confidence i.e. the best examples are at the top of the agree lists. So, each document is located at one position in the agree list of H1 and on another position in the agree list of H2. Based on these two positions in the different sorted agree lists, we reckon a new position, which is the average of the positions of the document d in both lists. Finally, we sort the documents according to their new average position, which is their new confidence value. Then, the best P positive and N negative examples are added to the labelled data set L depending on their new confidence value. This method is described in Algorithm 2 and is called the Balanced Merged Agreement Algorithm Using Documents Rank (BMAADR).

5. Experiments and Results

In this section, we present the results obtained by using the multi-view learning techniques, presented in section 4.2, combining high-level and low-level features. First, we will present the results obtained by the SAR algorithm, which will form the baseline and then compare these with those obtained with the proposed MAA, BMAA and BMAADR algorithms. All experiments are performed on a leave-one-out 5 cross validation basis with SVM classifiers. In particular, we use the SVMlight package[6] for classification and the MontyTagger of the MontyLingua package[7] [19] for part-of-speech tagging. In order to test models across domains, we propose to train different models based on one domain only at each time and test the classifiers over all domains together. So, each percentage can be expressed as the average results over all datasets.

5.1. The SAR algorithm

We first propose to show the results obtained with the SAR algorithm [1]. To perform the experiments, we used two views generated from a random split of low-level features together with maximum entropy

6 http://svmlight.joachims.org/

7 http://web.media.mit.edu/ hugo/montylingua/

Algorithm 2 The BMAADR Algorithm.

Input: L a set of labeled examples from one domain, U a set of unlabeled examples from another domain, $P = N = X$
Output: Trained classifier $H2$

$H1.AgreeList \leftarrow \{\}$
$H2.AgreeList \leftarrow \{\}$
for k iterations **do**
  Train a classifier $H1$ on view $V1$ of L
  Train a classifier $H2$ on view $V2$ of L
  Allow $H1$ and $H2$ to label U
  for all $d \in U$ **do**
    if $H1.Class[d] = H2.Class[d]$ **then**
      $H1.AgreeList \leftarrow AgreeList \cup \{< d; H1.Class[d] >\}$
      $H2.AgreeList \leftarrow AgreeList \cup \{< d; H2.Class[d] >\}$
    end if
  end for
  $Sort(H1.AgreeList, byDecrConf.)$
  $Sort(H2.AgreeList, byDecrConf.)$
  for all $d \in H1.AgreeList$ **do**
    $Rank_d = \frac{\sum_{k \in \{H1,H2\}} Rank_d^{kAgreeList}}{2}$
    $topAgreeList \leftarrow (d, Rank_d)$
  end for
  $L \leftarrow L \cup \{$Balanced P positive and N negative examples with the lower rank from $topAgreeList\}$
end for

classifiers with a unit variance Gaussian prior. Indeed, the actual implementation of SAR does not allow to test it with different views but only with random subsets of views, nor with different classifiers. The results are illustrated in Table 7.

Table 7. SAR Accuracy Results in Percentage.

	MPQA	RIMDB	CHES	WBLOG
Unigram	63.7	77.1	72.3	59.7
Bigram	59.8	65.2	64.9	62.2

The results show interesting properties. Models built upon unigrams mostly outperform models based on bigrams. One great advantage of only using low-level features is the ability to reproduce such experiments on different languages without further resources than just texts. However, a good training data set will have to be produced as the best results are obtained from the manually annotated corpus RIMDB with 77.1%.

5.2. The Merged Agreement Algorithms

In this section, we propose to compare the multi-view learning algorithms based on two views. The first view contains the seven high-level features expressed in section 2 and the second view is the set of unigrams or bigrams. As a consequence, we expect that the low-level classifier will gain from the agreements with the high-level classifier and will self-adapt to different new domains. In Table 8, we show the results obtained using unigrams as low-level features and in Table 9, the results using bigrams for each of the algorithms presented in section 4.2. Before explaining the results, we illustrate the behavior of each classifier in Figure 1 in terms of accuracy along the different iterations. The MAA classifier improves its accuracy just in the first few iterations and then starts to loose in accuracy. This is mainly due to the fact that the unbalanced labeled examples impair the performance. In this case, the average accuracy across domains reaches 75.6% in the best case, which is worse than the SAR best performance of 77.1%. However, the BMAA and BMAADR

show a different behavior as their accuracy decreases slightly between the sixth and eighth iteration and then remains almost constant for both classifiers. As such, they benefit of the agreement of both classifiers in the first iterations. The best accuracy is obtained by the BMAADR algorithm, which reaches an average accuracy of 80%, which outperforms SAR. It is also interesting to notice that in almost all cases, unigram low-level features provide better results than bigrams. The only exception is the RIMDB training set, where using bigrams as low-level features drastically improves the results compared to the unigram representation. Moreover, we see that automatically building a labelled data set, such as the WBLOG, can lead to interesting results as it shows the second best performance for unigrams, although it only presents the third best result for the bigram case.

Table 8. Accuracy Results for Unigrams in Percentage.

	MPQA	RIMDB	CHES	WBLOG
MAA	59.1	63.5	75.6	69.4
BMAA	59.4	65.2	79.5	69.7
BMAADR	59.4	65.4	80.0	69.9

Table 9. Accuracy Results for Bigrams in Percentage.

	MPQA	RIMDB	CHES	WBLOG
MAA	57.5	69.9	71.6	64.7
BMAA	57.5	76.6	77.9	65.2
BMAADR	57.5	76.7	77.2	65.5

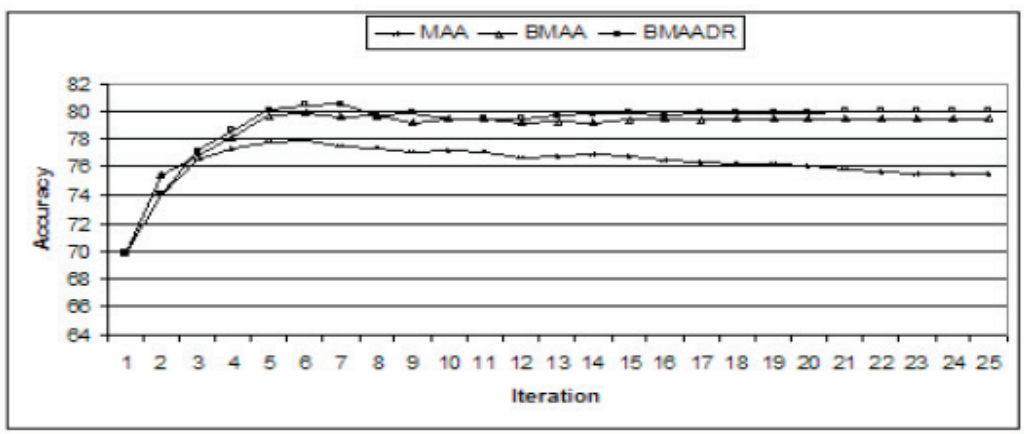

Fig. 1. Accuracies of Merged Algorithms.

The obtained results also show that SAR performs better in the cases of exclusively objective and subjective data sets (RIMDB and MPQA), while in the case of the other two data sets annotated at document level (i.e. texts do not contain exclusively objective or subjective sentences), the best classification accura-cies are obtained by the BMAADR. As a consequence, we can say that the BMAADR algorithm is the best performing algorithm for real-world texts situations. However, some comments must be claimed. In the proposed method, we rely on the assumption that the domain-independent view based on high-level features restricts the addition of wrongly predicted labels by both classifiers. However, these methods suffer of the weakness of the low-level classifier in its initial states, as wrong classifications may lead to produce small sets of examples, which may join the agree lists. Moreover, when both classifiers agree, they do not learn much more, especially if they agree with high-level of confidence in both classifiers. As a consequence, the

accuracy is almost constant for the models based on different views just after a few iterations. Nevertheless, we clearly believe that adapted multi-view or semi-supervised learning algorithms can lead to improve results compared to single-view approaches.

6. Conclusions

In this paper, we proposed to use a multi-view approach to address the problem of cross-domain sentiment classification. For that purpose, we presented three different algorithms based on an adaptation of the co-training algorithm by introducing different agreement constraints following the idea of [7]. The results showed the effectiveness of the proposed approach by combining high-level and low-level features as two different views. In particular, the best results showed accuracy of 80% across domains with the BMAADR algorithm compared to 77.1% for the SAR algorithm proposed by [1], the reference multi-view learning algorithm so far.

References

[1] K. Ganchev, J. Graca, J. Blitzer, B. Taskar, Multi-view learning over structured and non-identical outputs, in: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI 2008), 2008, pp. 204–211.
[2] A. Aue, M. Gamon, Customizing sentiment classifiers to new domains: a case study, in: Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP 2005), 2005, pp. 207–218.
[3] A. Finn, N. Kushmerick, Learning to classify documents according to genre, American Society for Information Science and Technology, Special issue on Computational Analysis of Style 57 (11) (2006) 1506–1518.
[4] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), 2007, pp. 187–205.
[5] E. Boiy, P. Hens, K. Deschacht, M.-F. Moens, Automatic sentiment analysis of on-line text, in: Proceedings of the 11th Interna-tional Conference on Electronic Publishing (ELPUB 2007), 2007, pp. 349–360.
[6] D. Lambov, G. Dias, V. Noncheva, Sentiment classification across domains, in: Proceedings of 14th Portuguese Conference in Artificial Intelligence (EPIA 2009), 2009.
[7] X. Wan, Co-training for cross-lingual sentiment classification, in: Proceedings of the Joint Conference of the 47th Annual Meet-ing of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL/IJCNLP 2009), 2009, pp. 235–243.
[8] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT 1998), 1998, pp. 92–100.
[9] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), 2004, pp. 271–278.
[10] C. Strapparava, R. Mihalcea, Learning to identify emotions in text, in: Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 2008), 2008, pp. 1556–1560.
[11] C. Strapparava, A. Valitutti, Wordnet-affect: An affective extension of wordnet, in: Proceedings of the 4th Language Resources and Evaluation International Conference (LREC 2004), 2004, pp. 1083–1086.
[12] V. Hatzivassiloglou, J. Wiebe, Effects of adjective orientation and gradability on sentence subjectivity, in: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), 2000, pp. 299–305.
[13] V. Hatzivassiloglou, K. McKeown, Predicting the semantic orientation of adjectives, in: Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics (EACL 1997), 1997, pp. 174–181.
[14] P. Chesley, B. Vincent, L. Xu, R. Srihari, Using verbs and adjectives to automatically classify blog sentiment, in: Proceedings of the AAAI Symposium on Computational Approaches to Analyzing Weblogs (AAAI/CAAW 2006), 2006, pp. 27–29.
[15] R. Srihari, W. Li, T. Cornell, C. Niu, Infoxtract: A customizable intermediate level information extraction engine, Natural Language Engineering (14) (2006) 33–69.
[16] B. Levin, English Verb Classes and Alternations, University of Chicago Press, 1993.
[17] G. A. Miller, Wordnet: an on-line lexical database, International Journal of Lexicography 3 (4).
[18] J. Rocchio, Relevance feedback in information retrieval, in: The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, 1971, Ch. 14, pp. 313–323.
[19] H. Liu, Montylingua: An end-to-end natural language processor with common sense (2004). URL http://web.media.mit.edu/hugo/montylingua