Grouping Product Features Using Semi-Supervised Learning with Soft-Constraints*

Zhongwu Zhai[†], **Bing Liu**[‡], **Hua Xu**[†] **and Peifa Jia**[†]

[†]State Key Lab of Intelligent Tech. & Sys.
Tsinghua National Lab for Info. Sci. and Tech.
Dept. of Comp. Sci. & Tech., Tsinghua Univ.
zhaizhongwu@gmail.com

[‡]Dept. of Comp. Sci.
University of Illinois at Chicago
liub@cs.uic.edu

Abstract

In opinion mining of product reviews, one often wants to produce a summary of opinions based on product features/attributes. However, for the same feature, people can express it with different words and phrases. To produce a meaningful summary, these words and phrases, which are domain synonyms, need to be grouped under the same feature group. This paper proposes a constrained semi-supervised learning method to solve the problem. Experimental results using reviews from five different domains show that the proposed method is competent for the task. It outperforms the original EM and the state-of-the-art existing methods by a large margin.

1 Introduction

One form of opinion mining in product reviews is to produce a feature-based summary (Hu and Liu, 2004a; Liu, 2010). In this model, product features are first identified, and positive and negative opinions on them are aggregated to produce a summary on the features. Features of a product are attributes, components and other aspects of the product, e.g., "picture quality", "battery life" and "zoom" of a digital camera.

In reviews (or any writings), people often use different words and phrases to describe the same product feature. For example, "picture" and "photo" refer to the same feature for cameras. Grouping such synonyms is critical for effective opinion summary. Although WorldNet and other thesaurus dictionaries can help to some extent, they are far from sufficient due to a few reasons. First, many words and phrases that are not synonyms in a dictionary may refer to the same feature in an application domain. For example, "appearance" and "design" are not synonymous, but they can indicate the same feature, *design*. Second, many synonyms are domain dependent. For example, "movie" and "picture" are synonyms in movie reviews, but they are not synonyms in camera reviews as "picture" is more likely to be synonymous to "photo" while "movie" to "video". Third, determining which expressions indicate the same feature can be dependent on the user's application need. For example, in car reviews, internal design and external design can be regarded as two separate features, but can also be regarded as one feature, called "design", based to the level of details that the user needs to study. In camera reviews, one may want to study battery as a whole (one feature), or as more than one feature, e.g., battery weight, and battery life. Due to this reason, in applications the user needs to be involved in synonym grouping.

Before going further, let us introduce two concepts, *feature group* and *feature expression*. Feature group (or *feature* for short) is the name of a feature (given by the user), while a feature expression of a feature is a word or phrase that actually appears in a review to indicate the feature. For example, a feature group could be named "picture quality", but there are many possible expressions indicating the feature, e.g., "picture", "photo", "image", and even the "picture quality" itself. All the feature expressions in a feature group signify the same feature.

Grouping feature expressions manually into suitable groups is time consuming as there are

*Supported by National Natural Science Foundation of China (Grant No: 60875073).

This work was done when the first author was visiting Bing Liu's group at the University of Illinois at Chicago.

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1272–1280,
Beijing, August 2010

often hundreds of feature expressions. This paper helps the user to perform the task more efficiently. To focus our research, we assume that feature expressions have been discovered from a review corpus by an existing system such as those in (Hu and Liu, 2004b; Popescu and Etzioni, 2005; Kim and Hovy, 2006; Kobayashi *et al.*, 2007; Mei *et al.*, 2007; Stoyanov and Cardie, 2008; Jin *et al.*, 2009; Ku *et al.*, 2009).

To reflect the user needs, he/she can manually label a small number of seeds for each feature group. The feature groups are also provided by the user based on his/her application needs. The system then assigns the rest of the feature expressions to suitable groups. To the best of our knowledge, this problem has not been studied in opinion mining (Pang and Lee, 2008).

The problem can be formulated as semi-supervised learning. The small set of seeds labeled by the user is the labeled data, and the rest of the discovered feature expressions are the unlabeled data. This is the transductive setting (Joachims, 1999) because the unlabeled set is used in learning and also in testing since our objective is to assign unlabeled expressions to the right feature groups.

Any semi-supervised learning method can be applied to tackle the problem. In this work, we use the Expectation-Maximization (EM) algorithm (Dempster *et al.*, 1977). Specifically, we use the naïve Bayesian EM formulation in (Nigam *et al.*, 2000), which runs a Bayesian classifier iteratively on the labeled and unlabeled data until the probabilities for the unlabeled data converge. When the algorithm ends, each unlabeled example is assigned a posterior probability of belonging to each group.

However, we can do better since the EM algorithm only achieves local optimal. What local optimal it achieves depends on the initialization, i.e., the initial seeds. We show that some prior knowledge can help provide a better initialization, and consequently generate better grouping results. Thus, we propose to create another set of data extracted from the unlabeled set based on two pieces of natural language knowledge:

1. Feature expressions sharing some common words are likely to belong to the same group, e.g., "battery life" and "battery power".
2. Feature expressions that are synonyms in a dictionary are likely to belong to the same group, e.g., "movie" and "picture".

We call these two pieces of prior knowledge *soft constraints* because they constrain the feature expressions to be in the same feature group. The constraints are soft (rather than hard) as they can be relaxed in the learning process. This relaxation is important because the above two constraints can result in wrong groupings. The EM algorithm is allowed to re-assign them to other groups in the learning process.

We call the proposed framework constrained semi-supervised learning. Since we use EM and soft constraints, we call the proposed method *SC-EM*. Clearly, the problem can also be attempted using some other techniques, e.g., topic modeling (e.g, LDA (Blei *et al.*, 2003)), or clustering using distributional similarity (Pereira *et al.*, 1993; Lin, 1998; Chen *et al.*, 2006; Sahami and Heilman, 2006). However, our results show that these methods do not perform as well.

The input to the proposed algorithm consists of: a set of reviews R, and a set of discovered feature expressions F from R (using an existing algorithm). The user labels a small set of feature expressions, i.e., assigning them to the user-specified feature groups. The system then assigns the rest of the discovered features to the feature groups. EM is run using the distributional (or surrounding words) contexts of feature expressions in review set R to build a naïve Bayesian classifier in each iteration.

Our evaluation was conducted using reviews from 5 different domains (insurance, mattress, vacuum, car and home-theater). The results show that the proposed method outperforms different variations of the topic modeling method LDA, k-means clustering, and the recent unsupervised feature grouping method mLSA.

In summary, this paper makes three main contributions:

1. It proposes a new sub-problem of opinion mining, i.e., grouping feature expressions in the context of semi-supervised learning. Although there are existing methods for solving the problem based on unsupervised learning, we argue that for practical use some form of supervision from the user is necessary to let the system know what the user wants.
2. An EM formulation is used to solve the problem. We augment EM with two soft constraints. These constraints help guide EM to

produce better solutions. We note that these constraints can be relaxed in the process to correct the imperfection of the constraints.

3. It is shown experimentally the new method outperforms the main existing state-of-the-art methods that can be applied to the task.

2 Related Work

This work is mainly related to existing research on synonyms grouping, which clusters words and phrases based on some form of similarity.

The methods for measuring word similarity can be classified into two main types (Agirre *et al.*, 2009): those relying on *pre-existing knowledge resources* (e.g., thesauri, or taxonomies) (Yang and Powers, 2005; Alvarez and Lim, 2007; Hughes and Ramage, 2007), and those based on *distributional properties* (Pereira *et al.*, 1993; Lin, 1998; Chen *et al.*, 2006; Sahami and Heilman, 2006; Pantel *et al.*, 2009).

In the category that relies on existing knowledge sources, the work of Carenini *et al.* (2005) is most related to ours. The authors proposed a method to map feature expressions to a given domain feature taxonomy, using several similarity metrics on WordNet. This work does not use the word distribution information, which is its main weakness because many expressions of the same feature are not synonyms in WordNet as they are domain/application dependent. Dictionaries do not contain domain specific knowledge, for which a domain corpus is needed.

Another related work is distributional similarity, i.e., words with similar meaning tend to appear in similar contexts (Harris, 1968). As such, it fetches the surrounding words as context for each term. Similarity measures such as *Cosine*, *Jaccard*, *Dice*, etc (Lee, 1999), can be employed to compute the similarities between the seeds and other feature expressions. To suit our need, we tested the *k*-means clustering with distributional similarity. However, it does not perform as well as the proposed method.

Recent work also applied topic modeling (e.g., LDA) to solve the problem. Guo *et al.* (2009) proposed a multilevel latent semantic association technique (called *mLSA*) to group product feature expressions, which runs LDA twice. However, *mLSA* is an unsupervised approach. For our evaluation, we still implemented the method and compared it with our SC-EM method.

Our work is also related to constrained clustering (Wagstaff *et al.*, 2001), which uses two forms of constraints, must-link and cannot-link. Must-links state that some data points must be in the same cluster, and cannot-links state that some data points cannot be in the same cluster. In (Andrzejewski *et al.*, 2009), the two constraints are added to LDA, called *DF-LDA*. We show that both these methods do not perform as well as our semi-supervised learning method *SC-EM*.

3 The Proposed Algorithm

Since our problem can be formulated as semi-supervised learning, we briefly describe the setting in our context. Given a set C of classes (our feature groups), we use L to denote the small set of labeled examples (labeled feature expressions or seeds), and U the set of unlabeled examples (unlabeled feature expressions). A classifier is built using L and U to classify every example in U to a class. Several existing algorithms can be applied. In this work, we use EM as it is efficient and it allows prior knowledge to be used easily. Below, we first introduce the EM algorithm that we use, and then present our augmented EM. The constraints and their conflict handling are discussed in Section 4.

3.1 Semi-Supervised Learning Using EM

EM is a popular iterative algorithm for maximum likelihood estimation in problems with missing data. In our case, the group memberships of the unlabeled expressions are considered missing because they come without group labels.

We use the EM algorithm based on naïve Bayesian classification (Nigam *et al.*, 2000). Although it is involved to derive, using it is simple. First, a classifier f is learned using only the labeled data L (Equations 1 and 2). Then, f is applied to assign a probabilistic label to each unlabeled example in U (see Equation 3). Next, a new classifier f is learned using both L and the newly probabilistically labeled unlabeled examples in U_{PL}, again using Equations 1 and 2. These last two steps iterate until convergence.

We now explain the notations in the Equations. Given a set of training documents D, each document d_i in D is considered as an ordered list of words. $w_{d_i,k}$ denotes the k^{th} word in d_i, where each word is from the vocabulary $V=\{w_1, w_2, \ldots, w_{|V|}\}$. $C=\{c_1, c_2, \ldots, c_{|C|}\}$ is the set of pre-defined

$$\mathrm{P}(w_t|c_j) = \frac{1 + \sum_{i=1}^{|D|} N_{ti}\mathrm{P}(c_j|d_i)}{|V| + \sum_{m=1}^{|V|}\sum_{i=1}^{|D|} N_{mi}\mathrm{P}(c_j|d_i)} \quad (1^1)$$

$$\mathrm{P}(c_j) = \frac{1 + \sum_{i=1}^{|D|} \mathrm{P}(c_j|d_i)}{|C| + |D|} \quad (2^1)$$

$$\mathrm{P}(c_j|d_i) = \frac{\mathrm{P}(c_j)\prod_{\mathrm{k}=1}^{|d_i|}\mathrm{P}(w_{d_i,k}|c_j)}{\sum_{\mathrm{r}=1}^{|C|}\mathrm{P}(c_r)\prod_{\mathrm{k}=1}^{|d_i|}\mathrm{P}(w_{d_i,k}|c_r)} \quad (3)$$

classes or groups. N_{ti} is the number of times the word w_t occurs in document d_i.

For our problem, the surrounding words contexts of the labeled seeds form L, while the surrounding words of the non-seed feature expressions form U. When EM converges, the classification labels of the unlabeled feature expressions give us the final grouping. Surrounding words contexts will be discussed in Section 5.

3.2 Proposed Soft-Constrained EM

Although EM can be directly applied to deal with our problem, we can do better. As we discussed earlier, EM only achieves local optimal based on the initialization, i.e., the labeled examples or seeds. We show that natural languages constraints can be used to provide a better initialization, i.e., to add more seeds that are likely to be correct, called *soft-labeled examples* or *soft seeds* (*SL*). Soft-labeled examples are handled differently from the original labeled examples in L. With the soft seeds, we have the proposed soft-constrained EM (called SC-EM).

Compared with the original EM, SC-EM has two main differences:

- Soft constraints are applied to L and U to produce a set *SL* of soft-labeled examples (or soft seeds) to initialize EM in addition to L. *SL* is thus a subset of U. The training set size is increased, which helps produce better results as our experimental results show.
- In the first iteration of EM, soft-labeled examples *SL* are treated in the same way as the labeled examples in L. Thus both *SL* and L are used as labeled examples to learn the initial classifier f_0. However, in the subsequent iterations, *SL* is treated in the same way as any examples in U. That is, the classifier f_x from each iteration x (including f_0) will predict U. After that, a new classifier is built using both L and U_{PL} (which is U with probabilistic labels). Clearly, this implies that the class labels of the examples in *SL* are allowed to change. That is also why we call *SL* the soft-labeled set in contrast to the hard-labeled set L, i.e., the examples in L will not change labels in EM. The reason that *SL* is allowed to change labels/classes is because the constraints can make mistakes. EM may be able to correct some of the mistakes.

[1] Laplace smoothing is used to prevent zero probabilities for infrequently occurring words.

Input:
- Labeled examples L
- Unlabeled examples U

1 Extract *SL* from U using constraints (Section 4);
2 Learn an initial naïve Bayesian classifier f_0 using $L \cup SL$ and Equations 1 and 2;
3 **repeat**
4 // E-Step
5 **for** each example d_i in U (including *SL*) **do**
6 Using the current classifier f_x to compute $\mathrm{P}(c_j|d_i)$ using Equation 3.
7 **end**
8 // M-Step
9 Learn a new naïve Bayesian classifier f_x from L and U by computing $\mathrm{P}(w_t|c_j)$ and $\mathrm{P}(c_j)$ using Equations 1 and 2.
10 **until** the classifier parameters stabilize

Output: the classifier f_x from the last iteration.

Figure 1. The proposed SC-EM algorithm

The detailed algorithm is given in Figure 1. The constraints are discussed in Section 4.

4 Generating SL Using Constraints

As mentioned earlier, two forms of constraints are used to induce the soft-labeled set *SL*. For easy reference, we reproduce them here:

1. Feature expressions sharing some common words are likely to belong to the same group.
2. Feature expressions that are synonyms in a dictionary are likely to belong to one group.

According to the number of words, feature expressions can be categorized into single-word expressions and phrase expressions. They are handled differently. The detailed algorithm is given in Figure 2. In the algorithm, L is the labeled set and U is the unlabeled set. L, in fact, consists of a set of sets, $L = \{L_1, L_2, \ldots, L_{|L|}\}$. Each L_i contains a set of labeled examples (feature expressions) of the i^{th} class (feature group). Similarly, the output set *SL* (the soft-labeled set) also consists of a set of sets, i.e., $SL = \{SL_1, SL_2, \ldots, SL_{|L|}\}$. Each SL_i is a set of soft-labeled examples (feature expressions) of the i^{th} class

(feature group). Thus L_i and SL_i correspond to each other as they represent the original labeled examples and the newly soft-labeled examples of the i^{th} class (or feature group) respectively.

The algorithm basically compares each feature expression u in U (line 1) with each feature expression e (line 4) in every labeled subset L_i (line 2) based on the above two constraints. If any of the constraints is satisfied (lines 5-17), it means that u is likely to belong to L_i (or the i^{th} class or feature group), and it is added to SL_i.

There are conflict situations that need to be resolved. That is, u may satisfy a constraint of more than one labeled sub-set L_i. For example, if u is a single word, it may be synonyms of feature expressions from more than one feature groups. The question is which group it is likely to belong. Further, u may be synonyms of a few single-word feature expressions in L_i. Clearly, u being a synonym of more than one word in L_i is better than it is only the synonym of one word in L_i. Similar problems also occur when u is an element of a feature expression phrase e.

To match u and e, there are a few possibilities. If both u and e are single words (lines 5-6), the algorithm checks if they are synonyms (line 7). The score in line 8 is discussed below. When one of u and e is a phrase, or both of them are phrases, we see whether they have shared words. Again, conflict situations can happen with multiple classes (feature groups) as discussed above. Note that in these cases, we do not use the synonym constraint, which does not help in our test.

Given these complex cases, we need to decide which class that u should be assigned to or should not be assigned to any class (as it does not meet any constraint). We use a score to record the level of satisfaction. Once u is compared with each e in every class, the accumulated score is used to determine which class L_i has the strongest association with u. The class j with the highest score is assigned to u. In other words, u is added to SL_j. Regarding the score value, synonyms gets the score of 1 (line 8), and intersection (shared words) gets the score equal to the size of the intersection (lines 10-17).

```
1  for each feature expression u ∈ U do
2     for each feature group L_i ∈ L do
3        score(L_i) ← 0;
4        for each feature expression e ∈ L_i do
5           if u is a single word expression then
6              if e is a single word expression then
7                 if u and e are synonyms then
8                    score(L_i) ← score(L_i) + 1;
9              else if w ∈ e then  // e is a phrase
10                score(L_i) ← score(L_i) + 1
11          else  // u is a phrase
12             if e is a single word expression then
13                if e ∈ u then  // u is a phrase
14                   score(L_i) ← score(L_i) + 1
15             else
16                s ← e ∩ u;
17                score(L_i) ← score(L_i) + |s|
18    u is added to SL_j s.t. argmax_{L_i} score(L_i)
```

Figure 2. Generating the soft-labeled set SL

5 Distributional Context Extraction

To apply the proposed algorithm, a document d_i needs to be prepared for each feature expression e_i for naïve Bayesian learning. d_i is formed by aggregating the distributional context of each sentence s_{ij} in our corpus that contains the expression e_i. The context of a sentence is the surrounding words of e_i in a text window of [-t, t], including the words in e_i. Given a relevant corpus R, the document d_i for each feature expression e_i in L (or U) is generated using the algorithm in Figure 3. Stopwords are removed.

```
1  for each feature expression e_i in L (or U) do
2     S_i ← all sentences containing e_i in R;
3     for each sentence s_ij ∈ S_i do
4        d_ij ← words in a window of [-t, t] on the left
                and right (including the words in e_i);
5     d_i ← words from all d_ij, j = 1, 2, …, |S_i|;
                // duplicates are kept as it is not union
```

Figure 3. Distributional context extraction

For example, a feature expression from L (or U) is e_i = "*screen*" and there are two sentences in our corpus R that contain "*screen*"

s_{i1} = "*The LCD **screen** gives clear picture*".

s_{i2} = "*The picture on the **screen** is blur*"

We use the window size of [-3, 3]. Sentence s_{i1}, gives us d_{i1} = <LCD, screen, give, clear, picture> as a bag of words. "the" and "is" are removed as stopwords. s_{i2} gives us d_{i2} = <picture, screen, blur>. "on", "the" and "is" are removed as stopwords. Finally, we obtain the document d_i for feature expression e_i as a bag of words:

d_i = <LCD, screen, give, clear, picture, picture, screen, blur>

6 Empirical Evaluation

This section evaluates the SC-EM algorithm and compares it with the main existing methods that can be applied to solve the problem.

6.1 Review Data Sets and Gold Standards

To demonstrate the generality of the proposed method, experiments were conducted using reviews from five domains: *Hometheater, Insurance, Mattress, Car* and *Vacuum*. All the data sets and the *gold standard* feature expressions and groups were from a company that provides opinion mining services. The details of the data sets and the gold standards are given in Table 1.

	Hometheater	Insurance	Mattress	Car	Vacuum
#Sentences	6355	12446	12107	9731	8785
#Reviews	587	2802	933	1486	551
#Feature expressions	237	148	333	317	266
#Feature groups	15	8	15	16	28

Table 1. Data sets and gold standards

6.2 Evaluation Measures

Since SC-EM is based on semi-supervised learning, we can use classification accuracy to evaluate it. We can also see it as clustering with initial seeds. Thus we also use clustering evaluation methods. Given gold standards, two popular clustering evaluation measures are *Entropy* and *Purity* (Liu, 2006). As *accuracy* is fairly standard, we will not discuss it further. Below, we briefly describe entropy and purity.

Given a data set *DS*, its gold partition is $G = \{g_1,\ldots, g_j,\ldots, g_k\}$, where k is the known number of clusters. The groups partition *DS* into k disjoint subsets, $DS_1,\ldots, DS_i, \ldots, DS_k$.

Entropy: For each resulting cluster, we can measure its entropy using Equation 4, where $\mathrm{P}_i(g_j)$ is the proportion of g_j data points in DS_i. The total entropy of the clustering (considering all clusters) is calculated by Equation 5.

$$entropy(DS_i) = -\sum_{j=1}^{k} \mathrm{P}_i(g_j) log_2 \mathrm{P}_i(g_j) \quad (4)$$

$$entropy_{total} = \sum_{i=1}^{k} \frac{|DS_i|}{|DS|} entropy(DS_i) \quad (5)$$

Purity: Purity measures the extent that a cluster contains only data from one gold-partition. Each cluster's purity is computed by Equation 6, and the total purity of the whole clustering is computed with Equation 7.

$$purity(DS_i) = \max_j \mathrm{P}_i(g_j) \quad (6)$$

$$purity_{total} = \sum_{i=1}^{k} \frac{|DS_i|}{|DS|} purity(DS_i) \quad (7)$$

In testing, the unlabeled set *U* is also our test set. This is justified because our purpose is to assign unlabeled data to appropriate groups.

6.3 Baseline Methods and Settings

The proposed ***SC-EM*** method is compared with a set of existing methods, which can be categorized into unsupervised and semi-supervised methods. We list the *unsupervised* methods first.

LDA: *LDA* is a popular topic modeling method (see Section 2). Given a set of documents, it outputs groups of terms of different topics. In our case, each feature expression is a term, and the documents refer to the distributional contexts of each feature expressions (see Section 5).

mLSA: This is a state-of-the-art unsupervised method for solving the problem. It is based on LDA, and has been discussed in related work.

Kmeans: This is the *k*-means clustering method (MacQueen, 1966) based on distributional similarity with cosine as the similarity measure.

In the *semi-supervised* category, the methods are further classified into un-constrained, hard-constrained, and soft-constrained methods.

For the *un-constrained* subclass (no constraints are used), we have the following:

LDA*(*L, H*)**: This method is based on ***LDA, but the labeled examples *L* are used as seeds for each group/topic. All examples in *L* will always stay in the same topic. We call this hard initialization (*H*). *L* is handled similarly below.

DF-LDA*(*L, H*)**. *DF-LDA* is the *LDA* method (Andrzejewski *et al.*, 2009) that takes must-links and cannot-links. Our *L* set can be expressed as a combination of must-links and cannot-links. Unfortunately, only must-links can be used because the number of cannot-links is huge and crashes the system. For example, for the car data, the number of cannot-links is 194,400 for 10% labeled data (see Section 6.4) and for 20% it is 466,560,000. DF-LDA also has a parameter η controlling the link strength, which is set very high (=1000) to reflect the hard initialization. We did not use *DF-LDA* in the unsupervised subclass above as without constraints it reduces to ***LDA.

***Kmeans*(*L, H*)**: This method is based on *Kmeans*, but the clusters of the labeled seeds are fixed at the initiation and remain unchanged.

***EM*(*L, H*)**: This is the original EM for semi-supervised learning. Only the labeled examples are used as the initial seeds.

For the *hard-constrained* (*H*) subclass (our

two constraints are applied and cannot be violated), we have the following methods (*LC* is *L* plus *SL* produced by the constraints (C):

***Rand*(*LC, H*):** This is an important baseline. It shows whether the constraints alone are sufficient to produce good results. That is, the final result is the expanded seeds *SL* plus the rest of *U* assigned randomly to different groups.

***LDA*(*LC, H*)**: It is similar to *LDA*(*L,H*), but both the initial seeds *L* and the expanded seeds *SL* are considered as labeled examples. They also stay in the same topics/groups in the process. Note that although *SL* is called a set of soft-labeled examples (seeds) in the proposed algorithm, they are treated as hard-labeled examples here just for experimental comparison.

***DF-LDA*(*LC, H*)**: This is *DF-LDA* with both *L* and *SL* expressed as must-links. Again, a large η (= 1000) is used to make sure that must-links for *L* and *SL* will not be violated.

***Kmeans*(*LC,H*)**: It is similar to *Kmeans*(*L,H*), but both *L* and *SL* stay in their assigned clusters.

***EM*(*LC, H*)**: It is similar to *SC-EM*, but *SL* is added to the labeled set *L*, and their classes are not allowed to change in the EM iterations.

For the *soft-constrained* (*S*) subclass, our two constraints can be violated. Initially, both the initial seeds *L* and the expanded seeds *SL* are considered as labeled data, but subsequently, only *L* is taken as the labeled data (i.e., staying in the same classes). The algorithm will re-estimate the label of each feature expression in *SL*. This subclass has the following methods:

***LDA*(*LC, S*)**: This is in contrast to *LDA*(*LC, H*). It allows the *SL* set to change topics/groups.

***Kmeans*(*LC, S*)**: This is in contrast to *Kmeans*(*LC, H*).

A soft *DF-LDA* is not included here because different η values give different results, and they are generally worse than *DF-LDA*(*LC, H*).

For all *LDA* based methods, the topic modeling parameters were set to their default values. The number of iteration is 1000. We used the *LDA* in MALLET[2], and modified it to suit different *LDA*-based methods except *DF-LDA*, which was downloaded from its authors' website[3]. We implemented *mLSA*, *Kmeans* and changed EM[4] to take soft seeds. For all *Kmeans* based methods, the distance function is the cosine similarity.

[2] http://mallet.cs.umass.edu/
[3] http://pages.cs.wisc.edu/~andrzeje/research/df_lda.html
[4] http://alias-i.com/lingpipe/

6.4 Evaluation Results

We now compare the results of *SC-EM* and the 14 baseline methods. To see the effects of different numbers of labeled examples (seeds), we experimented with 10%, 20%, 30%, 40%, and 50% of the feature expressions from the gold standard data as the labeled set *L*, and the rest as the unlabeled set *U*. All labeled data were selected randomly. For each setting, we run the algorithms 30 times and report the average results. Due to space limitations, we can only show the detailed *purity* (Pur), *entropy* (Ent) and *accuracy* (Acc) results for 30% as the labeled data (70% as unlabeled) in Table 2. For the other proportions of labeled data, we summarize them in Table 3. Each result in Table 3 is thus the average of the 5 data sets. All the results were obtained from the unlabeled set *U*, which was our test set. For entropy, the smaller the value is the better, but for purity and accuracy, the larger the better. For these experiments, we used the window size $t = 5$. Section 6.5 studies the effects of window sizes.

Tables 2 and 3 clearly show that the proposed algorithm (***SC-EM***) outperforms all 14 baseline methods by a large margin on every dataset. In detail, we observe the following:

- *LDA*, *mLSA* and *Kmeans* with no seeds (labeled data) perform the worst. Seeds help to improve the results, which is intuitive. Without seeds, *DF-LDA* is the same as *LDA*.
- *LDA* based methods seems to be the weakest. *Kmeans* based methods are slightly better, but EM based methods are the best. This clearly indicates that classification (EM) performs better than clustering. Comparing *DF-LDA* and *Kmeans*, their results are similar.
- For *LDA*, and *Kmeans*, hard-constrained methods (i.e., *LDA*(*L, H*), and *Kmeans*(*L, H*)) perform better than soft-constrained methods (i.e., *LDA*(*LC, S*) and *Kmeans*(*LC, S*)). This indicates that soft-constrained versions may change some correctly constrained expressions into wrong groups. However, for the EM based methods, the soft-constrained method (*SC-EM*) performs markedly better than the hard-constrained version (*EM*(*LC, H*)). This indicates that Bayesian classifier used in EM can take advantage of the soft constraints and correct some wrong assignments made by constraints. Much weaker results of *Rand*(*LC, H*) than *SC-EM* in different settings show that

Methods	Hometheater			Insurance			Mattress			Car			Vacuum		
	Acc	Pur	Ent	Acc	Pur	Ent	Acc	Pur	Ent	Acc	Pur	Ent	Acc	Pur	Ent
LDA	0.06	0.31	2.54	0.11	0.36	2.24	0.05	0.32	2.57	0.06	0.37	2.39	0.03	0.36	2.09
mLSA	0.06	0.31	2.53	0.14	0.38	2.19	0.06	0.34	2.55	0.09	0.37	2.40	0.03	0.37	2.11
Kmeans	**0.21**	**0.42**	**2.14**	**0.25**	**0.45**	**1.90**	**0.15**	**0.39**	**2.32**	**0.25**	**0.44**	**2.16**	**0.24**	**0.47**	**1.78**
LDA(L, H)	0.10	0.32	2.50	0.16	0.37	2.22	0.10	0.34	2.57	0.19	0.39	2.36	0.10	0.39	2.09
DF-LDA(L, H)	0.27	0.37	2.32	0.25	0.41	2.00	0.19	0.39	2.35	0.28	0.45	2.15	0.31	0.40	1.98
Kmeans(L, H)	0.20	0.42	2.12	0.25	0.43	1.92	0.17	0.42	2.26	0.27	0.48	2.04	0.20	0.48	1.76
EM(L, H)	**0.48**	**0.50**	**1.93**	**0.50**	**0.53**	**1.69**	**0.52**	**0.56**	**1.87**	**0.56**	**0.58**	**1.80**	**0.49**	**0.52**	**1.79**
Rand(CL, H)	0.41	0.46	2.07	0.40	0.46	1.94	0.40	0.47	2.07	0.34	0.41	2.31	0.39	0.52	1.59
LDA(CL, H)	0.44	0.50	1.96	0.42	0.48	1.89	0.42	0.49	1.97	0.44	0.52	1.87	0.43	0.55	1.48
DF-LDA(CL, H)	0.35	0.49	1.86	0.33	0.49	1.71	0.23	0.39	2.26	0.34	0.51	1.88	0.37	0.52	1.58
Kmeans(CL, H)	0.49	0.55	1.70	0.48	0.55	1.62	0.44	0.51	1.91	0.47	0.54	1.80	0.44	0.58	1.42
EM(CL, H)	**0.59**	**0.60**	**1.62**	**0.58**	**0.60**	**1.46**	**0.56**	**0.59**	**1.74**	**0.62**	**0.64**	**1.54**	**0.55**	**0.60**	**1.44**
LDA(CL, S)	0.24	0.35	2.44	0.27	0.40	2.14	0.23	0.37	2.44	0.27	0.41	2.33	0.23	0.41	2.01
Kmeans(CL, S)	0.33	0.46	2.04	0.34	0.45	1.90	0.25	0.43	2.20	0.29	0.47	2.07	0.37	0.50	1.68
SC-EM	**0.67**	**0.68**	**1.30**	**0.66**	**0.68**	**1.18**	**0.68**	**0.70**	**1.27**	**0.70**	**0.71**	**1.24**	**0.67**	**0.68**	**1.18**

Table 2. Comparison results (L = 30% of the gold standard data)

Methods	Acc					Pur					Ent				
	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
LDA	0.07	0.07	0.06	0.06	0.08	0.33	0.33	0.34	0.35	0.38	2.50	2.44	2.37	2.28	2.11
mLSA	0.07	0.07	0.08	0.07	0.07	0.34	0.35	0.35	0.37	0.38	2.48	2.42	2.36	2.26	2.12
Kmeans	**0.22**	**0.23**	**0.22**	**0.22**	**0.22**	**0.42**	**0.43**	**0.44**	**0.44**	**0.46**	**2.16**	**2.11**	**2.06**	**1.98**	**1.86**
LDA(L, H)	0.10	0.10	0.13	0.14	0.15	0.34	0.34	0.36	0.37	0.39	2.48	2.43	2.35	2.25	2.11
DF-LDA(L, H)	0.23	0.25	0.26	0.27	0.30	0.41	0.40	0.41	0.41	0.44	2.23	2.23	2.16	2.10	1.94
Kmeans(L, H)	0.13	0.16	0.22	0.24	0.28	0.42	0.43	0.45	0.45	0.48	2.15	2.11	2.02	1.95	1.79
EM(L, H)	**0.35**	**0.44**	**0.51**	**0.55**	**0.58**	**0.43**	**0.49**	**0.54**	**0.57**	**0.61**	**2.22**	**1.99**	**1.81**	**1.65**	**1.49**
Rand(CL, H)	0.28	0.35	0.39	0.42	0.45	0.39	0.43	0.47	0.50	0.54	2.33	2.15	2.00	1.82	1.63
LDA(CL, H)	0.31	0.38	0.43	0.46	0.49	0.43	0.47	0.51	0.54	0.58	2.16	1.99	1.83	1.69	1.49
DF-LDA(CL, H)	0.32	0.33	0.33	0.34	0.36	0.49	0.50	0.48	0.48	0.48	1.90	1.85	1.86	1.83	1.82
Kmeans(CL, H)	0.33	0.41	0.46	0.49	0.52	0.47	0.51	0.55	0.57	0.61	1.98	1.82	1.69	1.56	1.42
EM(CL, H)	**0.44**	**0.54**	**0.58**	**0.61**	**0.64**	**0.49**	**0.57**	**0.61**	**0.64**	**0.67**	**1.98**	**1.72**	**1.56**	**1.40**	**1.25**
LDA(CL, S)	0.17	0.21	0.25	0.30	0.34	0.34	0.36	0.39	0.42	0.46	2.47	2.37	2.27	2.09	1.87
Kmeans(CL, S)	0.23	0.28	0.32	0.36	0.42	0.43	0.44	0.46	0.48	0.51	2.15	2.08	1.98	1.86	1.70
SC-EM	**0.45**	**0.58**	**0.68**	**0.75**	**0.81**	**0.50**	**0.61**	**0.69**	**0.76**	**0.82**	**1.95**	**1.56**	**1.24**	**0.94**	**0.69**

Table 3. Influence of the seeds' proportion (which reflects the size of the labeled set L)

constraints alone (i.e., synonyms and sharing of words) are far from sufficient. EM can improve it considerably.

- Comparing EM based methods, we can see that soft seeds in *SL* make a big difference for all data sets. *SC-EM* is clearly the best.
- As the number of labeled examples increases (from 10% to 50%), the results improve for every method (except those for *DF-LDA*, which does not change much).

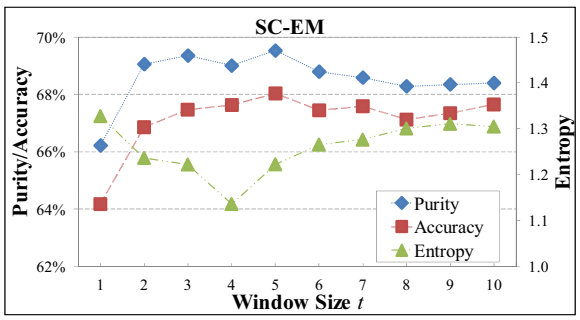

Figure 4. Influence of context window size

6.5 Varying the Context Window Size

We varied the text window size t from 1 to 10 to see how it impacts on the performance of *SC-EM*. The results are given in Figure 4 (they are averages of the 5 datasets). Again for purity and accuracy, the greater the value the better, while for entropy it is the opposite. It is clear that the window sizes of 2~6 produce similar good results. All evaluations reported above used t = 5.

7 Conclusion

This paper proposed the task of feature grouping in a semi-supervised setting. It argued that some form of supervision is needed for the problem because its solution depends on the user application needs. The paper then proposed to use the EM algorithm to solve the problem, which was improved by considering two soft constraints. Empirical evaluations using 5 real-life data sets show that the proposed method is superior to 14 baselines. In our future work, we will focus on further improving the accuracy.

References

Agirre E., E. Alfonseca, K. Hall, J. Kravalova, M. Pa ca and A. Soroa 2009. *A study on similarity and relatedness using distributional and WordNet-based approaches*. Proceedings of ACL.

Alvarez M. and S. Lim 2007. *A Graph Modeling of Semantic Similarity between Words*. Proceeding of the Conference on Semantic Computing.

Andrzejewski D., X. Zhu and M. Craven 2009. *Incorporating domain knowledge into topic modeling via Dirichlet forest priors*. Proceedings of ICML.

Blei D., A. Y. Ng and M. I. Jordan 2003. "*Latent Dirichlet Allocation*." JMLR 3: 993-1022.

Carenini G., R. Ng and E. Zwart 2005. *Extracting knowledge from evaluative text*. Proceedings of International Conference on Knowledge Capture.

Chen H., M. Lin and Y. Wei 2006. *Novel association measures using web search with double checking*. Proceedings of ACL.

Dempster A., N. Laird and D. Rubin 1977. "*Maximum likelihood from incomplete data via the EM algorithm*." Journal of the Royal Statistical Society 39(1): 1-38.

Guo H., H. Zhu, Z. Guo, X. Zhang and Z. Su 2009. *Product feature categorization with multilevel latent semantic association*. Proc. of CIKM.

Harris Z. S. 1968. *Mathematical structures of language*. New York, Interscience Publishers.

Hu M. and B. Liu 2004a. *Mining and summarizing customer reviews*. Proceedings of SIGKDD.

Hu M. and B. Liu 2004b. *Mining Opinion Features in Customer Reviews*. Proceedings of AAAI.

Hughes T. and D. Ramage 2007. *Lexical semantic relatedness with random graph walks*. EMNLP.

Jin W., H. Ho and R. Srihari 2009. *OpinionMiner: a novel machine learning system for web opinion mining and extraction*. Proceedings of KDD.

Joachims T. 1999. *Transductive inference for text classification using support vector machines*. Proceedings of ICML.

Kim S. and E. Hovy 2006. *Extracting opinions, opinion holders, and topics expressed in online news media text*. Proceedings of EMNLP.

Kobayashi N., K. Inui and Y. Matsumoto 2007. *Extracting aspect-evaluation and aspect-of relations in opinion mining*. Proceedings of EMNLP.

Ku L., H. Ho and H. Chen 2009. "*Opinion mining and relationship discovery using CopeOpi opinion analysis system*." Journal of the American Society for Information Science and Technology 60(7): 1486-1503.

Lee L. 1999. *Measures of distributional similarity*, Proceedings of ACL.

Lin D. 1998. *Automatic retrieval and clustering of similar words*, Proceedings of ACL.

Liu B. 2006. *Web data mining; Exploring hyperlinks, contents, and usage data*, Springer.

Liu B. 2010. *Sentiment Analysis and Subjectivity*. *Handbook of Natural Language Processing* N. Indurkhya and F. J. Damerau.

MacQueen J. 1966. *Some methods for classification and analysis of multivariate observations*. Proc. of Symposium on Mathematical Statistics and Probability.

Mei Q., X. Ling, M. Wondra, H. Su and C. Zhai 2007. *Topic sentiment mixture: modeling facets and opinions in weblogs*. Proceedings of WWW.

Nigam K., A. McCallum, S. Thrun and T. Mitchell 2000. "*Text classification from labeled and unlabeled documents using EM.*" Machine Learning 39(2).

Pang B. and L. Lee 2008. "*Opinion mining and sentiment analysis*." Foundations and Trends in Information Retrieval 2(1-2): 1-135.

Pantel P., E. Crestan, A. Borkovsky, A. Popescu and V. Vyas 2009. *Web-scale distributional similarity and entity set expansion*. EMNLP.

Pereira F., N. Tishby and L. Lee 1993. *Distributional clustering of English words*. Proceedings of ACL.

Popescu A.-M. and O. Etzioni 2005. *Extracting Product Features and Opinions from Reviews*. EMNLP.

Sahami M. and T. Heilman 2006. *A web-based kernel function for measuring the similarity of short text snippets*. Proceedings of WWW.

Stoyanov V. and C. Cardie 2008. *Topic identification for fine-grained opinion analysis*. COLING.

Wagstaff K., C. Cardie, S. Rogers and S. Schroedl 2001. *Constrained k-means clustering with background knowledge*. In Proceedings of ICML.

Yang D. and D. Powers 2005. *Measuring semantic similarity in the taxonomy of WordNet*, Proceedings of the Australasian conference on Computer Science.