A General Aspect-Term-Extraction Model for Multi-Criteria Recommendations

Paolo Pastore[1], Andrea Iovine[2], Fedelucio Narducci[1] and Giovanni Semeraro[2]

[1]Polytechnic University of Bari, Italy
[2]Dept. of Computer Science University of Bari, Italy

Abstract

In recent years, increasingly large quantities of user reviews have been made available by several e-commerce platforms. This content is very useful for recommender systems (RSs), since it reflects the users' opinion of the items regarding several aspects. In fact, they are especially valuable for RSs that are able to exploit multi-faceted user ratings. However, extracting aspect-based ratings from unstructured text is not a trivial task. Deep Learning models for aspect extraction have proven to be effective, but they need to be trained on large quantities of domain-specific data, which are not always available. In this paper, we explore the possibility of transferring knowledge across domains for automatically extracting aspects from user reviews, and its implications in terms of recommendation accuracy. We performed different experiments with several Deep Learning-based Aspect Term Extraction (ATE) techniques and Multi-Criteria recommendation algorithms. Results show that our framework is able to improve recommendation accuracy compared to several baselines based on single-criteria recommendation, despite the fact that no labeled data in the target domain was used when training the ATE model.

Keywords

multi-criteria recommendation, deep learning, aspect term extraction, domain adaptation, transfer learning

1. Introduction

Nowadays, many Web platforms and e-commerce websites allow customers to express their opinions by providing reviews on items, services, or media. Such *user-generated content* is extremely valuable for recommendation, since it reflects the user's perception of a specific item and of specific features of that item listing its strengths and weaknesses, the most important features, and the tasks for which it is more (or less) suitable. Extracting this information and exploiting it to enrich user profiles and item descriptions can give enormous advantages to Recommender Systems (RSs). Given the considerable importance of reviews in the recommendation process, many works in the literature proposed the idea of integrating them into RSs, as a way to improve their accuracy. Specifically, text reviews can be a solution to the *rating sparsity* problem often encountered by RSs based on Collaborative Filtering (CF), and can be used to capture a much more fine-grained model of the customer's preferences [1]. Accordingly, instead of modeling the user's profile as a set of *(item, rating)* pairs, it might be represented as a set of *(item, aspect, rating)* triples. Of course, the problem with this approach is that both aspects and ratings must be extracted automatically from *unstructured* text. This task is usually referred to as *Aspect-Based Sentiment Analysis* (ABSA). ABSA is not a trivial task, because there is no stable definition of "aspect", due to its intrinsic subjectivity. Also, the same aspect can appear in many different *forms* inside user reviews. For instance, a reviewer could use "service", "staff" or "waiter" for referring to the "service" category. For this reason, we distinguish between the aspect itself and its *representation forms* in the reviews, also called *aspect terms*. Furthermore, the aspects used in a domain are completely different to those in other domains: for restaurants, users will mention features such as the food or the quality of the service, when talking about smartphones, they will instead refer to other aspects such as the screen or the camera. In recent years, many models for automatically extracting aspects from text based on Deep Learning models have been proposed. However, these techniques need to be trained on domain-specific labeled datasets that are not always available.

In this paper, we investigate the application of domain adaptation strategies for aspect-based recommendation. The aim is to evaluate the effectiveness of modern Deep Learning-based Aspect Term Extraction (ATE) models when no annotated data is available for the target domain. For this purpose, we developed an aspect-based recommendation framework that includes an ATE module, an Aspect Clustering module, a Sentiment Analysis (SA) module, and a Multi-Criteria Recommender System. We performed an experimental study to compare several ATE models both in a single domain scenario and in a domain adaptation setting. We then chose the

3rd Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) & 5th Edition of Recommendation in Complex Environments (ComplexRec) Joint Workshop @ RecSys 2021, September 27–1 October 2021, Amsterdam, Netherlands
paolo.pastore1@poliba.it (P. Pastore); andrea.iovine@uniba.it (A. Iovine); fedelucio.narducci@poliba.it (F. Narducci); giovanni.semeraro@uniba.it (G. Semeraro)

model that obtained the best performance in both settings, i.e. the model that is most able to capture the essential, domain-invariant characteristics of aspect terms. Finally, we tested the framework in a recommendation scenario, to understand whether the models involved in this study actually improve the accuracy of RSs, compared to single-criteria recommendation baselines. This will prove that our framework is able to successfully extract fine-grained ratings from text, and exploit them for improving the quality of the recommendations.

In summary, the main contributions of this work are: (a) The definition of a novel framework for aspect-based recommendation, that can automatically extract aspect-based ratings from unstructured text (i.e. reviews) independently from the domain, using Deep Learning models; (b) An evaluation of the performance of Deep Learning-based ATE models in a domain adaptation setting (i.e. when no annotated data in the target domain is available); (c) An evaluation of the performance of our framework, compared to a set of single-criteria recommendation baselines, in terms of rating prediction accuracy.

2. Related work

A great amount of work has been dedicated to researching techniques for enhancing RSs by using data extracted from reviews. Chen et al. [1] and He et al. [2] contain a review of the state of the art of review-aware RSs. There are three main types of approaches: *Word-based*, that consists of directly using words found in the review as the user profile; *Sentiment-based*, that aims to extract the user's overall rating of an item via Sentiment Analysis; *Aspect-based*, that exploits multi-faceted ratings from reviews.Our work is strictly focused on aspect-based recommendation, extracting explicit factors from text reviews rather than latent factors (such as in [3, 4, 5]). The main advantage is that aspects can be also useful outside recommendation, e.g. for explanation.

Many works employ strategies such as topic modeling [6], sentiment lexicons [7], or rule-based systems [2, 8] in order to extract aspect-based ratings from reviews for recommendation purposes. The experiments performed in these works prove that aspect-based ratings can indeed improve recommendation accuracy over single-criteria baselines. In our work, we plan to instead perform the ATE task by using techniques based on Deep Learning.

In Musto et al. [9], ABSA is applied to a Multi-Criteria RS for the restaurant recommendation scenario using a tool called SABRE [10], which is able to extract relevant aspects from review text using the Kullback-Leibler divergence [11], as well as the rating assigned to each aspect. Aspects can also be organized into sub-aspects to obtain fine-grained information. Multi-criteria User-to-User and Item-to-Item CF algorithms were both proposed as recommendation algorithms. Our work follows a similar approach. In our framework however, the ATE task is performed using state-of-the-art Deep Learning models.

ABSA has proven to be a very effective method for improving the accuracy, usefulness and persuasiveness of the recommendations. As a result, Natural Language Processing (NLP) research focused on improving ABSA and ATE models, and more resources have been made available for these tasks. Examples of such resources are the SemEval datasets [12, 13, 14], and Hu and Liu [15].

Earlier works on ATE proposed strategies such as association rule mining [15], Conditional Random Fields (CRF) [16], knowledge-based topic modeling [17], or double propagation [18, 19]. In recent years, the success of Deep Learning models in Natural Language Processing tasks meant that research focus has moved towards using neural networks for ATE. Pavlopoulos and Androutsopoulos [20] improved the method described in [15] by using word embeddings generated via Word2Vec. Poria et al. [21] used Convolutional Neural Networks (CNNs) and several word embedding strategies. Giannakopoulos et al. [22] developed a model for both supervised and unsupervised ATE in large review datasets, based on Bi-Directional Long-Short Term Memory (Bi-LSTM) networks and CRF. Li and Lam [23] propose a multi-task learning framework for ATE and sentiment analysis based on LSTMs. Li et al. [24] use aspect detection history and opinion summary to enhance the ATE model. Some works investigate the addition of dependency relationships in order to improve the accuracy of neural network-based models, such as Ye et al. [25] and Luo et al. [26].

Finally, some works are focused on developing ATE methods that can generalize over different domains, using transfer learning or domain adaptation approaches. An early example is Jakob and Gurevych [16], which used a CRF-based approach. Ding et al. [27] use RNNs combined with rule-based auxiliary labels. Wang and Pan [28] incorporate dependency tree information using Recursive Neural Networks for both Aspect Term Extraction and Opinion Target Extraction tasks in order to transfer information between domains. Later, in [29] they introduce Transferable Interactive Memory Networks (TIMN) that can effectively model a representation for aspect terms across domains. Marcacini et al. [30] use transductive learning to map linguistic features of source and target domains in a heterogeneous network. Lee et al. [31] propose a transfer learning approach for ATE that is based on sequentially fine-tuning pre-trained features over different product groups. Pereg et al. [32] investigate the introduction of external syntactic features into a BERT-based model in order to exploit structural similarities of aspects across domains. Liang et al. [33] exploit the correlation between coarse-grained aspect categories and fine-grained aspect terms via a multi-level recon-

struction mechanism. In our work, we not only evaluate the performance of several ATE approaches in a domain adaptation setting, but we also assess their effectiveness in improving the accuracy of the recommendations.

Recently, Da'u et al. [34] investigated the application of Deep Learning aspect extraction models for recommendation. While this work has the same premise as ours, there are two major differences: first, the architecture used is based on CNNs, while we included several configurations based on residual LSTM and BERT. Second, their work relies on the presence of annotated ATE data for the target domain, and does not deal with domain adaptation.

Based on the analysis of the literature, we have identified a gap in the literature. In fact, the papers mentioned above either describe domain adaptation strategies for ATE, or employ ATE for recommendation purposes. To the best of our knowledge, none combine the two ideas together, by explicitly measuring the impact of domain adaptation on the quality of the recommendations. We believe that this is very important, especially due to the extreme scarcity of annotated datasets for training ATE systems, which hinders their applicability to the recommendation scenario.

3. Aspect-based recommendation framework

In this section, we describe a novel review-aware aspect-based recommendation framework that has been created for the purposes of this study. We exploit user reviews in order to go beyond item ratings, by extracting richer aspect-based evaluations. The main advantage of this framework is that it lets us discover new aspects directly from user reviews. Additionally, the aspect-based item ratings enrich the user profile, as they let us understand which aspects users care more about. Finally, they allow us to identify the individual strengths and weaknesses of each item from the user's point of view.

The proposed architecture is composed by several submodules as shown in the example in Figure 1. The first one is the ATE module which is in charge of identifying aspects mentioned in the user reviews, by extracting the corresponding aspect terms from the review text. The framework supports several ATE approaches, which will be detailed in Section 3.1.

The second component is the Aspect Clustering module, whose role is to group aspect terms that express similar concepts together into *aspects*. The Sentiment Analysis module works in parallel with the previous two. Its role is to extract the user's sentiment from the review in order to assign a score to each aspect term. Details on this step will be discussed in Section 3.2.

The outputs of the Aspect Clustering and Sentiment Analysis modules are used to compose the aspect-based item ratings, which are organized into a 3-dimesional tensor (i.e. a tensor in which the first dimension represents the users, the second represents the items, and the third represents the aspect clusters) which is then passed to the Multi-Criteria recommendation algorithm. More details on this component are discussed in Section 3.3.

Figure 1 shows an example of execution of our framework. Each review is split into atomic sentences, and then each sentence is given as input to both the ATE module and the SA module, in order to extract both aspect terms and ratings. In the example, starting from the sentence *"As always we had a great glass of wine while we waited"*, the ATE module extracts the *"glass of wine"* aspect term, and the SA module assigns a positive rating to it. The extracted aspect is then given as input to the Aspect Clustering module, that assigns it to the right cluster, i.e. *Beverage*. The cluster information and the predicted sentiments are used to generate the aspect-based ratings tensor. The Recommendation Algorithm takes this tensor as input for generating a list of recommendations.

3.1. Aspect Term Extraction

This section is focused on describing the ATE component of the framework. ATE is one of the sub-tasks of ABSA [14].

Most approaches treat the task of extracting relevant aspects as a sequence labeling problem [21], in which the review is first tokenized, and then each token is classified as either being an aspect term or not. A classifier can be trained by supplying supervised data, i.e. pre-annotated reviews. The standard schema for annotating reviews is the BIO tagging. According to this schema, three distinct labels can be associated to each token: B means that the token represents the beginning of an aspect term, I means that it represents the continuation of an aspect term, while O means that it is not an aspect term. This schema is shared with other sequence labeling tasks, such as Named Entity Recognition (NER).

Figure 2 shows the architecture of the ATE module. For this task, we focused on techniques based on Deep Learning, which have proven to be the most promising in the state of the art. In our study, we focused on the well known BERT model and on the residual Bi-LSTM. BERT is one of the most recent pre-trained frameworks for NLP and it can be exploited for many tasks, including NER and ATE. The residual Bi-LSTM is a variant of the classical Bi-directional LSTM which was successfully used in other sequence labeling tasks such as Tran et al. [35]. It is composed of two stacked Bi-LSTM layers, where the sum of the output of the first and second layer is sent to the final softmax layer, instead of sending only the output of the second layer. Different embedding strategies have

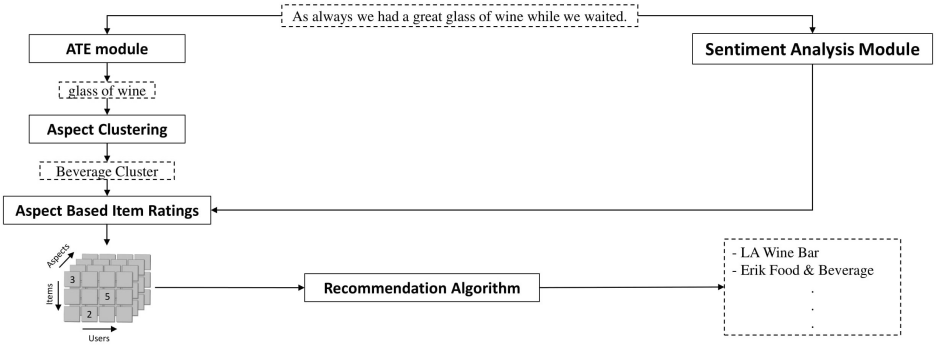

Figure 1: Example of recommendation process

been used in order to encode the tokens into real-valued vectors. In particular, we aim to use the ability to capture a contextual representation of words to learn a model that is *independent from the domain*, i.e. that is able to extract aspect terms from reviews of any domain. In this way, we can exploit a model trained on a given domain to extract aspect terms from another, unseen domain. Hence, the definition of *domain adaptation*.

The following is a list of all the ATE approaches that are included in the evaluation.

Pre-trained Word2Vec-Residual LSTM. Word2Vec is one of the first successful word embedding techniques, introduced in Mikolov et al. [36]. For this configuration, we employed embeddings that were previously trained from a part of the Google News datasets[1]. The neural network architecture used in this configuration is the Residual Bi-directional Long-Short Term Memory (LSTM) described earlier.

Pre-trained GloVe-Residual LSTM. For this approach, we used a set of pre-trained embeddings from GloVe. GloVe is a model for distributed word representation, introduced in Pennington et al. [37]. It is developed as an open-source project at Stanford University, and the pre-trained embeddings are publicly available[2]. The neural network architecture used is the Residual LSTM, like in the previous configurations.

ELMo embeddings-Residual LSTM. ELMo (Peters et al. [38]) stands for Embeddings from Language Models, and is a novel contextualized embedding strategy. That is, instead of using a single vector for each word in the dictionary, ELMo looks at the entire sentence before assigning each word in it its embedding. The result is that the embeddings generated by ELMo are deeply contextualized, and are more capable of handling polisemy. In this configuration, the architecture is defined as follows: an ELMo embedding layer is used, followed by the residual Bi-LSTM layers described in the previous configurations.

BERT. For this configuration, we employed BERT, introduced in Devlin et al. [39], which has been successfully applied in a variety of NLP tasks such as NER and text classification. Specifically, we employed a pre-trained BERT model available from the PyTorch library[3]. This model is then fine-tuned, i.e. its parameters are updated by training it on the ATE task. The NN architecture used by BERT is a multi-layer bidirectional Transformer encoder, as described in [39].

3.2. Aspect Term Clustering and Sentiment Analysis

As stated in the Introduction, one of the main problems of extracting aspect-based ratings from reviews is that users may refer to the same aspect in many different forms. Therefore, a strategy for grouping together all aspect forms that refer to the same concept is needed. We propose to group aspect terms together based on their Word2Vec representation. In the case of multi-word aspect terms, we calculated the average of the embeddings of each word. We then perform a clustering task by using the K-means algorithm. This allows us to automatically group aspect terms into aspect categories in an unsupervised way.

We then used the VADER sentiment analysis model offered by the NLTK library[4] to obtain the rating assigned to each aspect term in the review. Each review is split into atomic sentences, which are fed to the sentiment analyzer in order to predict their polarity. We then use this sentiment to assign a score to all the aspect terms

[1] https://code.google.com/archive/p/word2vec/?fbclid=IwAR3poHsG_4PZdqfbR_JESidu9WLMf44ffd0A8ZFmrxCPiKTDghc5hQCLUeQ

[2] https://nlp.stanford.edu/projects/glove/?fbclid=IwAR3JafEUyzBT5kwgdKHcQH20nQeTzG1NZs2_BHAhuOgaluO0HC7P5WW6EC8

[3] https://pypi.org/project/pytorch-pretrained-bert/

[4] https://www.nltk.org/

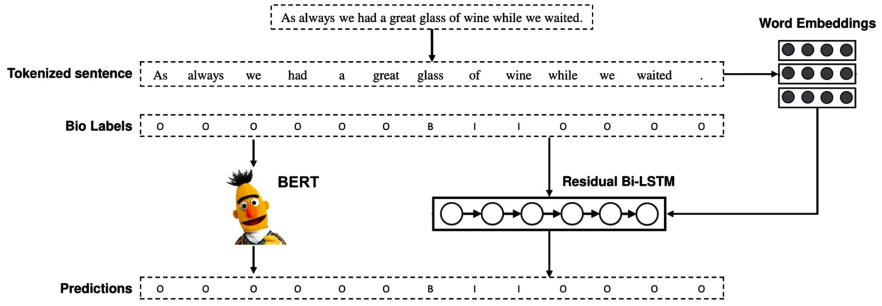

Figure 2: Execution of the ATE task with the residual Bi-LSTM and BERT

appearing in that sentence. The final output is the transformation of each review into a set of *(user, item, aspect, rating)* tuples. This information will be the input to the Multi-Criteria RS.

3.3. Aspect-Based Multi-Criteria recommendation

Once the proposed framework has extracted all aspect-based ratings from the reviews, the last step is the recommendation. Recommendations are generated via a multi-criteria algorithm based on collaborative filtering [40]. For this purpose, we treated the sentiments extracted by our framework as the ratings given by the user to the item for each aspect. For each aspect that was not mentioned in the user review, we decided to assign the item's overall rating. This choice was made empirically, as it improved the performance of the recommendation algorithm. The rest of this section contains a description of the recommendation algorithms.

User-to-User Multi-Criteria CF: This is an extension of the *similarity-based* approaches for CF. The distance $d(u_j, u_k)$ between users u_j and u_k is calculated using a multi-criteria distance function that takes the ratings given to each aspect into account (Equation 13 in [40]). For a new user-item pair, we generate a neighborhood of top-n most similar users, and then we calculate the predicted overall rating using the adjusted weighted sum of the neighbor's ratings (Equation 3 in [40]).

Item-to-Item Multi-Criteria CF: This is the multi-criteria equivalent of the item-based CF technique. As for the previous technique, the distance $d(i_j, i_k)$ between items is calculated using a multi-criteria distance function (Equation 5 in [9]). For any given user-item pair, we generate a neighborhood of the top-n most similar items. The overall predicted rating is calculated using the item-based equivalent of the adjusted weighted sum approach found in [40].

Multi-Criteria SVD: This approach is based on *Singular Value Decomposition* (SVD), which is a matrix factorization technique. More details about the SVD technique can be found in Koren et al. [41]. This technique was originally developed for single-criteria RSs. In order to extend it to a multi-criteria scenario, we used a naive *aggregation function-based* approach [40, 42]: we divided the k-dimensional multi-criteria recommendation task into a set of k single-criteria tasks. This means that we trained k SVD models, one for each aspect a_c, for $c \in \{1, ..., k\}$. Each model predicts the rating for a specific aspect $r_{a_c}(u, i)$. In order to predict the overall rating $r(u, i)$ for a given user u and an item i, we calculate an aggregate function: $r(u, i) = f(r_{a_1}(u, i), ..., r_{a_k}(u, i))$. In our case, the aggregate function is a simple average of the aspect-based ratings.

4. Evaluation

This section describes the in-vitro experiment that we set up to evaluate the performance of our framework. The experiment is divided into two parts. First, we evaluate the ATE models that were described in Section 3.1, in order to determine which one has the best performance when trained in a domain adaptation scenario. The second step of the experiment is the recommendation test: we extract aspect-based ratings from a dataset of restaurant reviews using the best ATE model from the previous test, and then we evaluate each of the multi-criteria recommendation approaches discussed in Section 3.3 in terms of their rating prediction accuracy. These approaches will also be compared to several baselines. This experiment will assess whether the multi-criteria recommendations generated by our framework are more accurate than the ones obtained by using single-criteria ratings.

4.1. Evaluation of the ATE approaches

We collected six datasets for the ATE task from the literature, three of which come from the SemEval ABSA

Table 1
Description of the datasets

Dataset	#Sentences	#Aspect terms
Restaurants (SemEval 2014-15-16)	7841	8183
Laptops (SemEval 2014)	3845	2918
Hotels (SemEval 2015)	266	213
Computers (Liu et al.)	531	363
Speakers (Liu et al.)	689	454
Routers (Liu et al.)	879	325

challenges with reviews about restaurants, laptops and hotels [12, 13, 14], while the other three are found in Liu et al. [18] and contain reviews about computers, speakers and routers. Table 1 reports the number of sentences and aspect terms contained in each dataset.

A *single domain* study was conducted by training and testing each ATE model on the same dataset. Train-test split was performed via 5-fold cross validation. The metrics used to evaluate the performance are Precision, Recall, and F1-score. An aspect term was considered correctly recognized if all the tokens that compose it were correctly tagged by the system. Therefore, partial matches were not considered in the evaluation. For each configuration, we calculated the overall score by averaging the metrics obtained for each fold.

In addition to the single domain study, we performed a *domain adaptation* experiment, which tests each model's ability to generalize the ATE task onto a new, unseen domain. We performed six tests, one for each dataset. In each test, we used one dataset as the test set, and all remaining datasets as the training and development set, using a random 80-20 split.

Table 2 describes the results of experiments. *Single* refers to the single domain tests, while *DA* refers to the domain adaptation tests. We report the Precision, Recall and F1-measure for each dataset and each model.

The table shows that the combination of ELMo embeddings with the residual Bi-LSTM is able to outperform all the other approaches, except for the domain adaptation scenario in the Laptop dataset, in which case BERT achieves slightly higher performance. Concerning the single domain experiment, it is also interesting to note that all four approaches perform better on the Restaurants dataset than on the Laptops dataset. This is not surprising, due to the fact that the Restaurants dataset is larger than the Laptops one. Even on the smaller datasets (Hotels, Speakers, Computers, Routers), ELMo still obtained the best performance.

However, the situation is less clear for the other approaches. On the Hotels dataset, which is the smallest one, GloVe and Word2Vec obtain second and third place, having a F1 of 0.612 and 0.528 respectively. BERT is again last, with 0.332, which may suggest that this approach is especially affected by training set size. An interesting observation can be made about the Routers dataset: despite not being the smallest dataset, all approaches performed especially poorly on it.

In the domain adaptation test, ELMo outperforms the other three models in five out of six datasets. We also compare the scores obtained from the single domain and domain transfer tests. In the largest datasets, we can observe that the latter induces a substantial loss in F1 compared to the former: around 28% in the Restaurants domain, and around 47% in the Laptops domain. This loss can be attributed to the lack of domain-specific data in the respective domains. In the smaller datasets such as Hotels, the loss is either very small, or nonexistent. Similar observations can be made for the BERT approach in the larger datasets. In the smaller datasets however, the domain transfer configuration actually outperforms the single domain one. This gives more credibility to the hypothesis that BERT is more susceptible to training set size compared to ELMo. The GloVe and Word2Vec approaches show much larger losses. This is a clear indication that they are less capable of transferring knowledge on the ATE task from one domain to another.

Based on the results from this Section, we can say with enough confidence that ELMo is the approach that obtained the best performance in the ATE task. Not only it outperformed the other three approaches in the single domain setting, but it is also demonstrated a good ability to transfer the aspect extraction task over different domains. For this reason, we chose this approach as part of the ATE component of our framework.

4.2. Evaluation of the Recommender System

We performed an experiment to measure our framework's recommendation accuracy. In particular, the objective of this experiment is to answer the following research questions:

RQ1: What is the impact of domain adaptation strategies for ATE on the quality of multi-criteria recommendations?

RQ2: How does our framework compare against several single-criteria baselines?

For this experiment, we employed the Yelp Recruiting Competition dataset[5], which contains restaurant reviews. This dataset is composed of 45, 981 users, 11, 537 items, and 229, 906 reviews, with a sparsity of around 99.95%. Each item in the dataset contains the user ID, the business ID, the review text, and an overall score given by the user on a 1-5 scale. The review set was also filtered by excluding all users that rated less than 10 items. The filtered dataset contains 4, 393 users, 10, 801 items, and 138, 301 reviews.

[5] https://www.kaggle.com/c/yelp-recruiting/data

Table 2
Results of the ATE task experiments

		Speakers				Computers				Routers			
		ELMo	BERT	GloVe	W2V	ELMo	BERT	GloVe	W2V	ELMo	BERT	GloVe	W2V
Single	P	0.682	0.372	0.486	0.452	0.506	0.334	0.448	0.462	0.462	0.24	0.424	0.24
	R	0.516	0.4	0.338	0.38	0.521	0.286	0.306	0.394	0.388	0.168	0.226	0.14
	F1	**0.576**	0.38	0.39	0.408	**0.514**	0.3	0.332	0.41	**0.406**	0.188	0.29	0.174
DA	P	0.55	0.412	0.17	0.146	0.61	0.46	0.31	0.258	0.39	0.276	0.084	0.048
	R	0.534	0.54	0.19	0.216	0.452	0.486	0.26	0.304	0.428	0.444	0.076	0.056
	F1	**0.534**	0.464	0.178	0.176	**0.52**	0.472	0.282	0.28	**0.408**	0.336	0.078	0.052

		Laptops				Hotels				Restaurants			
		ELMo	BERT	GloVe	W2V	ELMo	BERT	GloVe	W2V	ELMo	BERT	GloVe	W2V
Single	P	0.684	0.514	0.628	0.604	0.626	0.4	0.648	0.568	0.792	0.692	0.644	0.646
	R	0.68	0.514	0.622	0.632	0.63	0.308	0.596	0.5	0.784	0.706	0.642	0.638
	F1	**0.676**	0.51	0.626	0.618	**0.624**	0.332	0.612	0.528	**0.784**	0.696	0.642	0.638
DA	P	0.508	0.436	0.092	0.08	0.648	0.592	0.61	0.542	0.67	0.59	0.186	0.186
	R	0.282	0.31	0.04	0.046	0.624	0.672	0.552	0.464	0.496	0.364	0.096	0.096
	F1	0.358	**0.36**	0.056	0.06	**0.632**	0.628	0.578	0.5	**0.564**	0.444	0.126	0.126

4.2.1. Experimental protocol

The dataset was input to our framework, and all the steps described in Section 3 were performed. Aspect terms were extracted by using the ELMo approach. For this experiment, we used two ATE models: one trained on all six datasets described in Section 4.1, and another was trained without the Restaurants datasets, which allows us to assess the difference in recommendation quality caused by the lack of annotated ATE training data in the target domain.

The aspect terms were then grouped together into k aspects, and ratings were assigned via the Sentiment Analysis component described in Section 3.2, which transformed each review into a $k+1$-dimensional vector, containing the user's rating of the restaurant for each of the k aspects, plus the overall rating. We experimented with different sizes of k (10, 30 and 50) in order to increase the generality of the results. Finally, the aspect-based rating vectors were passed to the recommendation algorithms described in section 3.3. We evaluated the rating prediction accuracy of the algorithms by measuring the Mean Average Error (MAE). 10-fold cross-validation was performed on the dataset, and the MAE values for each fold were averaged together. For each of the three multi-criteria recommendation algorithms (User-to-user, Item-to-item, and SVD), we chose the combination of parameters that obtained the best results. These models were then compared against several baselines: single-criteria user-to-user CF (with MSD and Pearson similarity measures), single-criteria item-to-item CF (with MSD and Pearson similarity measures), Singular Value Decomposition (SVD), and Non-negative Matrix Factorization (NMF), which were also trained and tested using 10-fold cross-validation. For both user-to-user and item-to-item CF baselines, we employed the variants that take into account the user and item means, to make them more comparable with the multi-criteria equivalents. This lets us understand whether the aspect-based ratings extracted by our framework actually cause an improvement in recommendation accuracy.

4.2.2. Results

Table 3 reports the results obtained by the three multi-criteria recommendation algorithms supported by our framework, with different combinations of parameters. For the user-to-user and item-to-item algorithms, we chose to set the neighborhood size to 10, 20, 30, 80, and 200. We chose these numbers as using a higher number of neighbors caused a decrease in the accuracy. For all three algorithms, we can observe that the best performance is obtained by using 10 aspects. This means that by increasing the number of aspects, the performance decreases. This makes sense, since the effectiveness of the multi-criteria distance metrics largely depend on the number of commonly rated aspects between the two users (or the two items). Increasing the number of aspects also increases the sparsity of the aspect-based ratings, which makes these metrics less effective. Table 3 shows that the multi-criteria user-to-user algorithm performs best by setting the neighborhood size to 200, with a MAE of 0.8147 and 0.8155 respectively for the model trained with and without the Restaurants dataset. For the multi-criteria item-to-item variant, the best neighborhood size is 80 for the model trained with the Restaurants dataset, and 200 for the model trained without it. In both the neighborhood-based models, we can observe that the model trained without the Restaurants dataset performs slightly worse than the one trained with all datasets. This

Table 3
Results for the Multi-Criteria algorithms (MAE). The best results for each algorithm are in italic. The best overall results are in bold.

		10 Aspects		30 Aspects		50 Aspects	
Algorithm	**#N.**	**W/Rest.**	**W/O Rest.**	**W/Rest.**	**W/O Rest.**	**W/Rest.**	**W/O Rest.**
M.C. U2U	10	0.83	0.8306	0.8314	0.8333	0.8329	0.8349
M.C. U2U	20	0.8196	0.8206	0.821	0.8228	0.8222	0.8244
M.C. U2U	30	0.8169	0.8178	0.8182	0.8199	0.8194	0.8214
M.C. U2U	80	0.8148	0.8157	0.8161	0.8176	0.8172	0.8191
M.C. U2U	200	*0.8147*	*0.8155*	0.8159	0.8174	0.817	0.8189
M.C. I2I	10	0.831	0.8321	0.8333	0.8346	0.8347	0.8364
M.C. I2I	20	0.8221	0.8228	0.8239	0.8252	0.8252	0.8269
M.C. I2I	30	0.82	0.8206	0.8216	0.8229	0.8228	0.8246
M.C. I2I	80	*0.8183*	0.819	0.8199	0.8211	0.8211	0.8227
M.C. I2I	200	0.8184	*0.8189*	0.8199	0.8211	0.8211	0.8227
M.C. SVD	-	***0.8062***	***0.8053***	0.8064	0.8069	0.8074	0.8081

Table 4
Results of the recommendation test. Best results are in bold.

Configuration	**MAE**
M.C. U2U (W/ Rest.)	0.8147
M.C. U2U (W/O Rest.)	0.8155
U2U (MSD)	0.8169
U2U (Pearson)	0.8565
M.C. I2I (W/ Rest.)	0.8183
M.C. I2I (W/O Rest.)	0.8189
I2I (MSD)	0.8202
I2I (Pearson)	0.8582
M.C. SVD (W/ Rest.)	0.8062
M.C. SVD (W/O Rest.)	**0.8053**
SVD	0.8107
NMF	0.8737

is consistent with the observations made during the experiment described in section 4.1, i.e. the loss in recommendation accuracy may be caused by a loss in ATE accuracy. However, this is not true the multi-criteria SVD approach. In fact, the model trained without the Restaurants dataset achieved better performance (MAE: 0.8053) compared to the one trained on all datasets (MAE: 0.8062). This suggests that this approach is less susceptible to the aspect-based rating sparsity problem. A Wilcoxon test was performed to evaluate the significance of these differences. The test confirms that they are all significant ($p < 0.01$). We can answer RQ1 by stating that that the proposed domain adaptation strategy for ATE does indeed cause a sensible loss in recommendation performance in the multi-criteria user-to-user and item-to-item algorithms. However, it also was associated to an equally small increase in the multi-criteria SVD algorithm.

Finally, in Table 4 we compare the performance of our framework with the baselines described earlier. We evaluated the single-criteria user-to-user and item-to-item baselines by setting the neighborhood size to 10, 20, 30, 80, and 200, and reported the best performance for each baseline. The results show that all three multi-criteria algorithms are able to outperform their single-criteria equivalents. The best result overall is achieved by the multi-criteria SVD on the model trained without restaurants. In fact, even though it is based on a basic aggregation function-based approach, it managed to obtain a significant improvement over all baselines. A Wilcoxon statistical test was performed in order to verify the significance of the difference in MAE. The test was able to prove that indeed the multi-criteria SVD approach performed significantly better than all the baselines with $p < 0.01$. This allows us to confidently answer RQ2 by stating that our framework compares favorably against all the selected baselines even when no domain-specific ATE data was available during training. This proves that the proposed domain adaptation approach is able to effectively exploit review data in order to improve the recommendation accuracy.

5. Conclusion

In this paper, we presented an investigation on the use of domain adaptation strategies in order to perform Aspect Term Extraction without the need for domain-specific training data, as well as the impact of using this strategy in a multi-criteria recommender system. For this purpose, we developed an aspect-based recommendation framework that automatically extracts multi-criteria ratings from text reviews using state-of-the-art Deep Learning ATE models. We performed several experiments to evaluate the ATE component both in a single domain and in a domain adaptation setting in order to find the best model to use in the multi-criteria recommendation scenario. We trained the aspect term extraction component twice: with domain-specific data, and without domain-specific data, and tested several combinations of parameters and different multi-criteria recommendation algorithms in order to increase the generality of the results. In all cases, the framework was able to outperform single-criteria baselines, with small differences between the two models. Moreover, the proposed strategy improves the quality of the recommendations even when no domain-specific ATE training data is available.

The most important limitation to the validity of our experiment is related to the small amount of data available for the ATE task. However, it is worth noting that this is a limitation of the state of the art, since all works on the subject use the same datasets (or a subset of them) that we used in our work. As future work, we plan to extend this work by including more recent Deep Learning architectures for ATE. We also plan to extend the recommendation test, by including more multi-criteria recommendation algorithms, and by comparing our framework with systems that extract latent factors from reviews.

References

[1] L. Chen, G. Chen, F. Wang, Recommender systems based on user reviews: the state of the art, User Modeling and User-Adapted Interaction 25 (2015) 99–154. URL: http://link.springer.com/10.1007/s11257-015-9155-5. doi:10.1007/s11257-015-9155-5.

[2] X. He, T. Chen, M.-Y. Kan, X. Chen, TriRank: Review-aware Explainable Recommendation by Modeling Aspects, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15, ACM Press, Melbourne, Australia, 2015, pp. 1661–1670. URL: http://dl.acm.org/citation.cfm?doid=2806416.2806504. doi:10.1145/2806416.2806504.

[3] R. Catherine, W. Cohen, Transnets: Learning to transform for recommendation, in: Proceedings of the eleventh ACM conference on recommender systems, 2017, pp. 288–296.

[4] S. Seo, J. Huang, H. Yang, Y. Liu, Representation learning of users and items for review rating prediction using attention-based convolutional neural network, in: International Workshop on Machine Learning Methods for Recommender Systems, 2017.

[5] P. Li, A. Tuzhilin, Latent multi-criteria ratings for recommendations, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 428–431.

[6] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS), in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14, ACM Press, New York, New York, USA, 2014, pp. 193–202. URL: http://dl.acm.org/citation.cfm?doid=2623330.2623758. doi:10.1145/2623330.2623758.

[7] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, S. Ma, Explicit factor models for explainable recommendation based on phrase-level sentiment analysis, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14, ACM Press, Gold Coast, Queensland, Australia, 2014, pp. 83–92. URL: http://dl.acm.org/citation.cfm?doid=2600428.2609579. doi:10.1145/2600428.2609579.

[8] K. Bauman, B. Liu, A. Tuzhilin, Recommending Items with Conditions Enhancing User Experiences Based on Sentiment Analysis of Reviews., in: CBRecSys@ RecSys, 2016, pp. 19–22.

[9] C. Musto, M. de Gemmis, G. Semeraro, P. Lops, A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews, in: Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17, ACM Press, Como, Italy, 2017, pp. 321–325. URL: http://dl.acm.org/citation.cfm?doid=3109859.3109905. doi:10.1145/3109859.3109905.

[10] A. Caputo, P. Basile, M. de Gemmis, P. Lops, G. Semeraro, G. Rossiello, SABRE: A Sentiment Aspect-Based Retrieval Engine, in: C. Lai, A. Giuliani, G. Semeraro (Eds.), Information Filtering and Retrieval: DART 2014: Revised and Invited Papers, Studies in Computational Intelligence, Springer International Publishing, Cham, 2017, pp. 63–78. URL: https://doi.org/10.1007/978-3-319-46135-9_4. doi:10.1007/978-3-319-46135-9_4.

[11] J. M. Joyce, Kullback-Leibler Divergence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 720–722. URL: https://doi.org/10.1007/978-3-642-04898-2_327. doi:10.1007/978-3-642-04898-2_327.

[12] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, SemEval-2014 Task 4: Aspect Based Sentiment Analysis (2014) 9.

[13] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 2015, pp. 486–495.

[14] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, SemEval-2016 Task 5: Aspect Based Sentiment Analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp. 19–30.

[15] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, Association for Computing Machinery, Seattle, WA, USA, 2004, pp. 168–177. URL: https://doi.org/10.1145/1014052.1014073. doi:10.1145/1014052.1014073.

[16] N. Jakob, I. Gurevych, Extracting opinion targets in a single-and cross-domain setting with conditional random fields, in: Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics, 2010, pp. 1035–1045.

[17] Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, R. Ghosh, Exploiting domain knowledge in aspect extraction, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1655–1667.

[18] Q. Liu, Z. Gao, B. Liu, Y. Zhang, Automated rule selection for aspect extraction in opinion mining, in: Twenty-Fourth International Joint Conference

on Artificial Intelligence, 2015.

[19] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, Z. Gao, Improving opinion aspect extraction using semantic similarity and aspect associations, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[20] J. Pavlopoulos, I. Androutsopoulos, Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method, in: Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM), Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 44–52. URL: http://aclweb.org/anthology/W14-1306. doi:10.3115/v1/W14-1306.

[21] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowledge-Based Systems 108 (2016) 42–49. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705116301721. doi:10.1016/j.knosys.2016.06.009.

[22] A. Giannakopoulos, C. Musat, A. Hossmann, M. Baeriswyl, Unsupervised Aspect Term Extraction with B-LSTM & CRF using Automatically Labelled Datasets, in: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2017, pp. 180–188.

[23] X. Li, W. Lam, Deep Multi-Task Learning for Aspect Term Extraction with Memory Interaction, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2886–2892. URL: http://aclweb.org/anthology/D17-1310. doi:10.18653/v1/D17-1310.

[24] X. Li, L. Bing, P. Li, W. Lam, Z. Yang, Aspect term extraction with history attention and selective transformation, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 4194–4200.

[25] H. Ye, Z. Yan, Z. Luo, W. Chao, Dependency-Tree Based Convolutional Neural Networks for Aspect Term Extraction, in: J. Kim, K. Shim, L. Cao, J.-G. Lee, X. Lin, Y.-S. Moon (Eds.), Advances in Knowledge Discovery and Data Mining, volume 10235, Springer International Publishing, Cham, 2017, pp. 350–362. URL: http://link.springer.com/10.1007/978-3-319-57529-2_28. doi:10.1007/978-3-319-57529-2_28, series Title: Lecture Notes in Computer Science.

[26] H. Luo, T. Li, B. Liu, B. Wang, H. Unger, Improving aspect term extraction with bidirectional dependency tree representation, IEEE/ACM Transactions on Audio, Speech, and Language Processing 27 (2019) 1201–1212. Publisher: IEEE.

[27] Y. Ding, J. Yu, J. Jiang, Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[28] W. Wang, S. J. Pan, Recursive Neural Structural Correspondence Network for Cross-domain Aspect and Opinion Co-Extraction, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2171–2181. URL: http://aclweb.org/anthology/P18-1202. doi:10.18653/v1/P18-1202.

[29] W. Wang, S. J. Pan, Transferable interactive memory network for domain adaptation in fine-grained opinion extraction, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 7192–7199. Issue: 01.

[30] R. M. Marcacini, R. G. Rossi, I. P. Matsuno, S. O. Rezende, Cross-domain aspect extraction for sentiment analysis: A transductive learning approach, Decision Support Systems 114 (2018) 70–80. URL: http://www.sciencedirect.com/science/article/pii/S0167923618301386. doi:10.1016/j.dss.2018.08.009.

[31] Y. Lee, M. Chung, S. Cho, J. Choi, Extraction of Product Evaluation Factors with a Convolutional Neural Network and Transfer Learning, Neural Processing Letters 50 (2019) 149–164. URL: https://doi.org/10.1007/s11063-018-9964-8. doi:10.1007/s11063-018-9964-8.

[32] O. Pereg, D. Korat, M. Wasserblat, Syntactically Aware Cross-Domain Aspect and Opinion Terms Extraction, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 1772–1777. URL: https://www.aclweb.org/anthology/2020.coling-main.158. doi:10.18653/v1/2020.coling-main.158.

[33] T. Liang, W. Wang, F. Lv, Weakly Supervised Domain Adaptation for Aspect Extraction via Multi-level Interaction Transfer, IEEE Transactions on Neural Networks and Learning Systems (2021). Publisher: IEEE.

[34] A. Da'u, N. Salim, I. Rabiu, A. Osman, Recommendation system exploiting aspect-based opinion mining with deep learning method, Information Sciences 512 (2020) 1279–1292. Publisher: Elsevier.

[35] Q. Tran, A. MacKinlay, A. J. Yepes, Named Entity Recognition with stack residual LSTM and trainable bias decoding, arXiv:1706.07598 [cs] (2017). URL: http://arxiv.org/abs/1706.07598, arXiv: 1706.07598.

[36] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, arXiv:1301.3781 [cs] (2013). URL: http:

//arxiv.org/abs/1301.3781, arXiv: 1301.3781.

[37] J. Pennington, R. Socher, C. Manning, Glove: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. URL: https://www.aclweb.org/anthology/D14-1162. doi:10.3115/v1/D14-1162.

[38] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv:1802.05365 [cs] (2018). URL: http://arxiv.org/abs/1802.05365, arXiv: 1802.05365.

[39] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805 [cs] (2019). URL: http://arxiv.org/abs/1810.04805, arXiv: 1810.04805.

[40] G. Adomavicius, Y. Kwon, New Recommendation Techniques for Multicriteria Rating Systems, IEEE Intelligent Systems 22 (2007) 48–55. doi:10.1109/MIS.2007.58.

[41] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37. Publisher: IEEE.

[42] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender Systems Handbook, Springer US, Boston, MA, 2011. URL: http://link.springer.com/10.1007/978-0-387-85820-3. doi:10.1007/978-0-387-85820-3.