



An Entity Based LDA for Generating Sentiment Enhanced Business and Customer Profiles from Online Reviews

Aniruddha Tamhane, Divyaa L. R., and Nargis Pervin^(✉)

Indian Institute of Technology, Madras, Chennai, India
nargisp@iitm.ac.in

Abstract. The accelerated growth of the Web2.0 has led to an abundance of accessible information which has been successfully harnessed by many researchers for personalizing products and services. Many personalization algorithms are focused on analyzing only the explicitly provided information and this limits the scope for a deeper understanding of the individuals' preferences. However, analyzing the reviews posted by the users seeks to provide a better understanding of users' personal preferences and also aids in uncovering business' strengths and weaknesses as perceived by the users. Topic Modeling, a popular machine learning technique addresses this issue by extracting the underlying abstract topics in the textual data. In this study, we present entity-LDA (eLDA), a variation of Latent Dirichlet Allocation for topic modeling along with a dependency tree based aspect level sentiment analysis methodology for constructing user and business profiles. We conduct several experiments for evaluating the quantitative and qualitative performance of our proposed model compared to state-of-the-art methods. Experimental results demonstrate the efficacy of our proposed method both in terms topic quality and interpretability. Finally we develop a framework for constructing user and business profiles from the topic probabilities. Further we enhance the business profiles by extracting syntactic aspect level sentiments to indicate sentimental polarity for each aspects.

Keywords: Topic modeling · Latent Dirichlet Allocation · Profiling Personalization · Sentiment analysis

1 Introduction

The digital revolution has led to an exponential increase in the information accessible online, which has found ample utility in personalization of products and services. By providing a more relevant and engaging experience to the users with personalized content, a significant increase in revenue and market share has been achieved. Many businesses like Yelp, Amazon, Netflix, etc. have their core business models centered around product personalization. Specifically, in the recommendation arena of movies, articles, books, etc. personalization is indispensable [23].

A vast majority of personalization techniques have focused on extracting explicit information on users and businesses through online ratings issued by a user. For example, Collaborative Filtering (CF) based algorithms personalize product recommendations based on the users' and items' historical ratings [16]. Social network based personalization algorithms further enhance the CF approaches by incorporating the user's social network related information. On the other hand, Content-based algorithms incorporate attributes extracted from the business itself to generate personalized recommendation. However, these approaches fail to capture user's inherent preferences and the ground-level facts about the businesses, which are better reflected in the reviews written by the users. Research on extracting information on user-business interactions through online reviews has focused on a variety of topic modeling and sentiment analysis approaches to gain insight into the user's perspective and opinions [4, 5, 37]. However, lack of generalization and difficulty in interpretability by humans have limited the scope of their practical applications.

In this work, we propose a topic modeling based sentiment enhanced entity (user/item) profiling methodology that generates user and item profiles as numeric vectors from the review text. The proposed framework consists of three stages: text preprocessing, topic modeling, and profile generation. In text preprocessing, we adapt a codeword insertion step to replace topic relevant terms with the domain specific category names based on the business' features. For example, in the context of restaurant reviews, the words "pizza" and "pepperoni" would be replaced by "American" if the restaurant serves North American cuisines and would be replaced by "Continental" if the restaurant serves Italian cuisines instead. We further parse the text to get Parts of Speech (POS) tags and a dependency tree in order to extract nouns for topic modeling and aspect level dependencies for sentiment classification.

In the topic modeling stage, we aggregate all the nouns and noun phrases from the preprocessed reviews at entity-level and generate two separate corpora, corresponding to user-level and business-level documents. We implement the entity Latent Dirichlet Allocation (eLDA) for extracting the topics, where "entity" refers to a user or item to be profiled. The derived topics are validated using several quantitative and qualitative (human judgment-based) metrics. The extracted topics are then mapped to the profile aspects in the profile generation stage. This ensures a domain-specific interpretability of the profiles. In addition, we use an aspect level sentiment classifier to classify the business aspects into strength/weaknesses as perceived by users. Finally, the entity profile vectors are generated as an aggregation of the probabilities of all the topics related to the aspects.

For this study, we have considered restaurant reviews posted by users in a publicly available dataset released by Yelp for Yelp dataset challenge to generate personalized profiles for users and restaurants. Our proposed method outperforms the benchmarked algorithms both in quantitative and qualitative metrics. This can be attributed to the following aspects of the proposed approach:

Noun Extraction: Extracting nouns and noun phrases retains the information critical to generating the profile, while ignoring the irrelevant information [22]. This in turn improves the topic quality and also reduces the computational time for topic modeling technique.

Codeword Insertion: Replacing multiple words referring to the same object with a codeword explicitly indicating the category reinforces domain-specific connections among words, that are not apparent from the corpus. Replacing redundant terms with codewords reduces the vocabulary size without loss of information. This enhances topic interpretability by concentrating the word-topic probability over a fewer set of words. It also improves discoverability of implicit topics that are not directly aggregated with a codeword.

The topic probabilities obtained from the proposed model are then used for generating entity profiles as a numeric vector of pre-determined dimension using the online reviews. Furthermore, we have enhanced the business profile by classifying the business aspects into user-perceived strengths and weaknesses using the parse tree based aspect level sentiment analysis.

The rest of the paper has been structured as follows: Sect. 2 presents literature related to this work. The solution overview has been described in detail in Sect. 3 followed by dataset description and experimental findings in Sect. 4 Sect. 5, respectively. Finally, conclusion and future work has been presented in Sect. 7.

2 Literature Review

In this section, we have reviewed the following topics relevant to our work: (1) Topic Modeling (2) Sentiment Analysis (3) Profiling and Personalization.

2.1 Topic Modeling

Information retrieval from massive text corpora has attracted attention from various research communities over the years. A plethora of work has focused on reducing the dimensionality of the word-frequency space by clustering various documents based on common underlying topics. The traditional Term Frequency-Inverse Document Frequency (TF-IDF) approach proposed by Salton and Buckley [28] reduces a document of arbitrary length to a real vector having a dimension equal to the total number of terms in the vocabulary of the entire corpus. Each vector element represents the term frequency (TF) weighted by the logarithm of inverse of the proportion of documents containing that term (IDF). Clustering algorithms are then applied to the TF-IDF matrix to cluster documents. This approach is appealing because of its simplicity and intuitive interpretability; however, it fails to capture underlying topics in the document, which can be modeled based on the co-occurrence of multiple terms. This problem has been addressed by Hoffman in the probabilistic Latent Semantic Indexing (pLSI) approach [12]. pLSI models a document as a probabilistic mixture of multiple topics, where a topic itself is a probabilistic distribution over the

terms in the vocabulary. However, this method is not scalable since the number of parameters grow linearly with the number of documents in the corpus and lacks generalized application in other corpus.

The Latent Dirichlet Allocation (LDA) approach proposed by Blei et al. [5] addresses these issues by including a document level multinomial probability distribution over topics in the pLSI model. This ensures that the learnt parameters are not document specific, do not grow linearly with the number of documents and can be used to apply topics to another corpus built from the same vocabulary. LDA has been widely applied in a number of domains, successfully yielding topics of good quality. Variations of LDA such as local-LDA by Brody and Eldadah [6] and sentence LDA (sent-LDA) by Bao and Dutta [1] have been developed to further improve topic quality in specific domains. Another probabilistic topic modeling approach, Correlated Topic Models (CTM) has been developed by Blei and Lafferty [4] to model and extract topics having a high correlation. CTM supports the extraction of a greater number of topics from the same corpus with a higher log-likelihood; however, CTM is computationally more expensive as compared to LDA, and has a comparable performance for a lower number of topics.

The approaches discussed so far have the ability to soft cluster groups of words into various topics based on their co-occurrence in the corpus. However, they do not have any means of enforcing a domain-specific understanding of words that may not be captured in a bag of words approach because of the inherent richness of the domain's vocabulary. Bao et al proposed an LDA based topic modeling approach where ratings corresponding to the reviews were incorporated in the topic modeling algorithm in [2]. This approach improved recommendation quality, but failed to explicitly generate topics and entity profiles. A semantic approach to topic modeling has been proposed by Linshi [17], where words of similar sentiment polarity have been grouped to reveal the sentiment associated with the individual topics. This approach has demonstrated the potential for revealing additional information through the incorporation of semantic knowledge in topic modeling. In this study, we have adapted a semantic-LDA hybrid approach by incorporating domain-specific knowledge in topic modeling, resulting in better topic quality and interpretability.

2.2 Sentiment Analysis

Sentiment Analysis (SA) is the computational treatment of subjective opinions and emotions expressed by people in texts. SA has most commonly been approached as a classification problem [24], where entire documents or parts of documents are classified to be associated with various sentiments. SA is done on three levels: document-level, sentence-level and aspect-level. Document-level SA classifies a document into a multitude of sentiments such as positive, negative, neutral, etc. Sentence-level SA makes the same classification for individual sentences instead of documents and the aspect-level SA associates different sentiments to various aspects of the document [24].

Aspect-level SA has the potential to provide better insight into the distinct polarity associated with various themes in rich review texts compared to other techniques where an aggregated polarity is given. It primarily involves three major steps: aspect detection, sentiment classification and aggregation [29]. Aspect detection methods have generally been frequency based [13, 18], syntactic [26, 38] or machine learning based [14, 19]. Frequency based and syntactic approaches are intuitive and have similar performance to machine learning techniques [29]. The sentiment classification, the second step, can occur separately or jointly [37] with aspect detection using either a lexicon-based bag of words approach for sentiment aggregation or implementing algorithms such as LDA, logistic regression, Hidden Markov Models, K-Nearest Neighbour, and Support Vector Machines for supervised and semi-supervised learning of the sentiments [24]. Aggregation of the sentiment associated with an aspect is the weighted average of all the individual sentiments [33, 34]. In this study, we have employed a hybrid aspect-level SA approach to extract the sentiment level polarities associated with the aspects of business profiles. A parse-tree based syntactic approach has been used for aspect detection and a machine learning based supervised classification approach for sentiment classification. Finally, we have aggregated the sentiments associated with an aspect by taking a simple average of the classification probabilities.

2.3 Profiling and Personalization

The exponential growth of the Internet has led to abundance of information at the disposal of web users. Personalization has helped tackle information overload by limiting nonessential content. For catering personalized services, accurately profiling users and businesses has become paramount. This has opened a new area of research for many researchers to mine both implicit (visits, clicks, time spent, etc.) and explicit (reviews, stars, etc.) information for profiling and personalization. Collaborative Filtering (CF) is a technique used for personalizing content by grouping similar users and businesses based on historical ratings [16, 27]. The underlying assumption is users who rated similarly in the past tend to rate similar in the future. One main limitation of similarity based CF is the time complexity does not grow linearly with number of users and businesses. Rather than exploiting the entire user space to find similar users, researchers have started using online social networks for identifying users' preferences. In [15, 20, 35] the latent features of friends from online social networks are assumed to be similar to target users' latent features and the influence from friends' networks are utilized for personalizing content to the target user. Although personalization can be facilitated by incorporating recommendations from friends in social network, there is a potential in incorporating individual preferences and perspectives about the businesses for quality profiling. In order to maintain individual preferences of experienced users along with social network influence Feng and Qian in [11] proposed a framework that fuses personal interest along with social influence. The personal interest factor captures the most desirable business categories for the target user by analyzing the historical ratings. The recent

growth of GPS enabled gadgets and Web 2.0 Technologies have attracted users to update their location information via check-ins. Prior works in [7, 39–41] focused on only geographical influence for providing recommendations whereas recent works in [9, 36] uses both geographical and social influences for providing Point-Of-Interest (POI) recommendations. Though the previously stated works provide personalized information using explicit information (ratings or stars), they do not analyze the user preferences that are latent in the form of reviews, posts, blogs, etc.

In this study, we propose an elegant approach to generate profiles for users and items by incorporating latent features mined from the user reviews using codeword based entity-level LDA technique. Further, sentiment analysis technique has been employed to detect sentiments associated with the topics.

3 Solution Overview

3.1 Problem Formulation

In this study, we aim to construct M user and N item profiles as k -dimensional real-valued vectors $U_u \in \mathbb{R}^k \forall u \in \mathbb{U}$ and $I_i \in \mathbb{R}^k \forall i \in \mathbb{I}$ from review texts, where \mathbb{U} is the set of users and \mathbb{I} is the set of items, such that $|\mathbb{U}| = M$ and $|\mathbb{I}| = N$.

3.2 Solution Details

Herein, we propose a Latent Dirichlet Allocation (LDA) based entity profiling algorithm (*eLDA*) that ensures that the domain specific user interests are captured in user profile $U_u \forall u \in \mathbb{U}$. Item profiles $I_i \forall i \in \mathbb{I}$ captures the key business attributes along with the sentiment polarity which indicates the strengths and weaknesses as perceived by the users in general. The overall architecture of *eLDA* has been presented in Fig. 1.

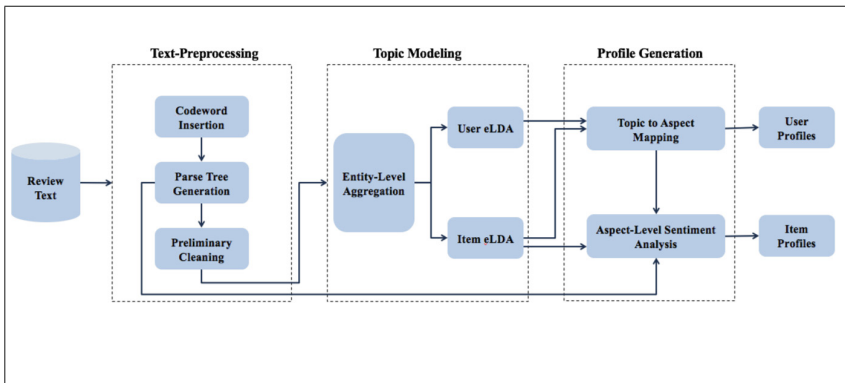


Fig. 1. Architecture for entity profile generation

Our algorithm consists of three major stages: text pre-processing, topic modeling, and entity profile generation. Each step has been explained in details below.

Text Pre-processing. Text pre-processing is customary in any Natural Language Processing technique. In eLDA this process consists of three stages, namely, codeword insertion, parse tree generation, and preliminary cleaning.

Codeword Insertion. We aggregate multiple words that represent the same category in the texts by replacing them with a common “codeword”. For example, the words ice-cream, cake, gateau, pastry are replaced by the codeword “desserts”. This work is focused on constructing restaurant profiles on a publicly available dataset collected from Yelp. To effectively tag the codewords, bag-of-words dictionary \mathbb{B} corresponding to each codeword (category) has been generated. Instead of manually creating the codeword dictionary, crawling multiple recipe, cuisine and professional restaurant review websites with appropriately categorized recipe names would be a convenient route. As a result of this data collection process, our bag-of-words dictionary contains over 5000 words corresponding to 10 distinct cuisine categories, namely: American, Continental, Southern, Alcohol, Desserts, Asian, Indo-Arabic, Sandwiches and Breakfast-Brunch. We use the bag-of-words list to replace the words with the corresponding codeword, determined by the context or the business features F_i . In our case $|F| = 10$. For example, we have replaced the word “pizza” with “American” if the restaurant serves North American cuisine, and “Continental” if it serves Italian cuisines. Thus, we define the codeword insertion operator $C(v, \mathbb{B})$ for word v and dictionary \mathbb{B} as:

$$C(v, \mathbb{B}) = \begin{cases} \mathbb{B}[v] & \mathbb{B}[v] \in F_i \\ v & \mathbb{B}[v] \notin F_i \end{cases} \quad (1)$$

Additionally, to detect if a restaurant enjoys location advantage, location specific keywords such as address of the restaurant has been added in a location codeword dictionary. Each dictionary category label is used as the respective codeword.

Parse Tree Generation. After codeword insertion, we parse the texts in order to generate the parse trees along with Parts Of Speech (POS) tags associated with the words. The POS tags are used to extract the nouns and noun phrases, which are later used in the topic modeling stage, to gain maximum critical information [22]. The parse trees are used for the aspect-level sentiment analysis of the business features in the profile generation stage. We have used the Compositional Vector Grammar based Parser [21, 32] available as a part of the Stanford NLP package for generating parse trees in our dataset. The parse tree attaches a POS tag t and an index j denoting the tree-node position for each of the words obtained from the codeword transformation. Thus, the tree transformation converts each word v into a triplet:

$$T(v) = (v, t, j) \quad (2)$$

Here, $T(v)[i]$ corresponds to the word v , POS tag t and tree-node position j for $i = 1, 2, 3$, respectively.

Preliminary Cleaning. After generating parse trees, we remove the stop-words, punctuations and special characters in order to retain information critical to topic modeling and sentiment analysis. Finally, we lemmatize and stem the words to their root words in order to improve topic quality and interpretability [3].

Topic Modeling. We formulate the entity Latent Dirichlet Allocation (eLDA) algorithm as an extension of LDA [5] for extracting topics to construct entity-level profiles described as follows:

1. Extract nouns and noun phrases i.e., aggregate that all the $w = C(\mathbb{B}, v)$ s.t. $T(w)[2] = \text{NN}$ or NNS or NNP or NNPS after codeword insertion from each of the reviews.
2. Generate M user-level aggregated documents $E_u \forall u \in \mathbb{U}$ and N item-level aggregated documents $E_i \forall i \in \mathbb{I}$ s.t. E_e is the aggregation of all the pre-processed nouns present in the reviews pertaining to the entity $e \in \mathbb{U}$ or $e \in \mathbb{I}$.
3. For each entity aggregation E_e :
 - (a) Choose $L \sim \text{Poisson}(\xi)$
 - (b) Choose $\theta \sim \text{Dirichlet}(\alpha)$
 - (c) For each of the L nouns in E_e :
 - i. Choose a topic z_l from Multinomial (θ).
 - ii. Choose a noun w_n from $p(w_n|z_n, \beta)$, a multinomial probability distribution

We obtain the probability of $\mathbf{w}_u, \mathbf{w}_i$ (which are the term frequency forms of E_u and E_i), the user-level corpus D_u and item-level corpus D_i of entity-level aggregations as:

$$p(\mathbf{w}_u | \alpha, \beta) = \int p(\theta_u | \alpha) \left(\prod_{l=1}^{L_u} \sum_{z_{u_l}} p(z_{u_l} | \theta_u) p(w_{u_l} | z_{u_l}, \beta) \right) d\theta_u \quad (3)$$

$$p(\mathbf{w}_i | \alpha, \beta) = \int p(\theta_i | \alpha) \left(\prod_{l=1}^{L_i} \sum_{z_{i_l}} p(z_{i_l} | \theta_i) p(w_{i_l} | z_{i_l}, \beta) \right) d\theta_i \quad (4)$$

$$p(D_u | \alpha, \beta) = \prod_{u \in \mathbb{U}} \int p(\theta_u | \alpha) \left(\prod_{l=1}^{L_u} \sum_{z_{u_l}} p(z_{u_l} | \theta_u) p(w_{u_l} | z_{u_l}, \beta) \right) d\theta_u \quad (5)$$

$$p(D_i | \alpha, \beta) = \prod_{i \in \mathbb{I}} \int p(\theta_i | \alpha) \left(\prod_{l=1}^{L_i} \sum_{z_{i_l}} p(z_{i_l} | \theta_i) p(w_{i_l} | z_{i_l}, \beta) \right) d\theta_i \quad (6)$$

Finally, the joint probability of a topic allocation θ_e to an entity aggregation E_e is given by:

$$p(\theta_e, \mathbf{z}_e | \mathbf{w}_e, \alpha, \beta) = \frac{p(\theta_e, \mathbf{z}_e, \mathbf{w}_e | \alpha, \beta)}{p(\mathbf{w}_e, \alpha, \beta)} \quad (7)$$

We use Gibbs sampling for estimating the optimal parameters for inferring the topic distribution since the integral in Eq. 3 is generally hard to compute. Finally, we extract a set of topics T_u, T_i from document corpora D_u, D_i and the associated document-topic probability matrices P_u, P_i .

3.3 Entity Profile Generation

The entity profile generation stage is executed in three steps: topic to aspect mapping, aspect-level sentiment analysis, and profile generation.

Topic to Aspect Mapping. The user-level topics extracted from *eLDA* for the user-level document corpus D_u and business-level document corpus D_i are mapped to k domain-specific profile categories where the mapping is defined as:

$$\mathcal{M} : \{T_u, T_i\} \rightarrow \Pi \quad (8)$$

Here, Π refers to the domain specific set of categories. In general, $|\Pi| = k \leq \max|T_u|, |T_i|$.

Aspect Level Sentiment Analysis. We use the parse tree transformation $T(v, t, j)$ generated in the pre-processing step for aspect identification and the corresponding sentiment classification. We use a supervised classifier \mathbb{K} trained on a sentiment-labeled review dataset for a binary (positive/negative) classification of the sentiments. It is worthy to note that we do not apply sentiment classification for *User eLDA* as that would misrepresent the user profiles. We argue that if a user is concerned about a particular aspect of a restaurant, then only he/she mentions that in a review. Hence, the sentiment towards the identified aspects will always be positive.

The aspect level sentiment analysis proceeds as follows:

For every item level document $d_i \in D_i \ \forall i \in \mathbb{I}$:

1. Aspects $A_i \leftarrow \{v | t = \text{NN or NNS or NNP or NNPS} \ \forall T(v) = (v, t, j)\}$
2. $temp \leftarrow \{\}$
3. For each of the aspects v in A_i :
 - (a) Generate the dependency text δ as the aggregation of the children of v , i.e. aggregate all u s.t. $j' = T(u, t', j') [3] \in \text{children}[j]$
 - (b) After preliminary cleaning mentioned in Sect. 3.2, generate the Document-Term frequency matrix dtm_i for δ .
 - (c) Use classifier \mathbb{K} to obtain probability of positive sentiment $s_i = \mathbb{K}(dtm_i)$
 - (d) Store s_i corresponding to v in $temp$ s.t. $temp[v] = s_i$
4. Use Eq. 8 to map individual aspects in A_i to the k profile categories and aggregate the sentiment probabilities for all aspects under each category as an arithmetic mean. Thus, $temp, \mathcal{M} \rightarrow S_i$
5. Store S_i in S .

Profile Generation. We aggregate the topic probabilities P_u and P_i using the mapping \mathcal{M} given in Eq. 8 to obtain the k -dimensional profile category probabilities \mathbb{P}_u and \mathbb{P}_i .

The User Profile U_u is defined as a k -dimensional numerical vector where the k elements of the vector denote the probability that the user u is interested in the each of the k categories in Π . Thus,

$$U_u = [\mathbb{P}_{ur}]_{k \times 1} \quad (9)$$

where \mathbb{P}_{ur} is the probability that user u is interested in category Π_r .

The business profile I_i has been characterized in analogy to the user profile, with the additional incorporation of the normalized sentiment score S_i . The normalized sentiment score includes a sentiment polarity with the existing profile topics enabling to capture the positive and negative aspects. Thus, the business profile is defined as:

$$I_i = [\mathbb{P}_{is} \times f_{is}]_{k \times 1} \quad (10)$$

where \mathbb{P}_{is} is the probability that business i is associated with category Π_s and S_{is} is the overall sentiment of business i with respect to category Π_s . Also, f_{is} is the sentiment polarity of item i with respect to category s defined as:

$$f_{is} = \begin{cases} -1 & S_{is} \in [0, l_1] \\ 1 & S_{is} \in [l_1, l_2] \\ 2 & S_{is} \in [l_2, 1] \end{cases} \quad (11)$$

Here, l_1 and l_2 are the experimentally decided cut-offs for the negative and neutral sentiments. Thus, a sentiment polarity of -1 denotes a negative sentiment, 1 denotes neutral sentiment and 2 denotes a positive sentiment.

4 Dataset Description

In this study, we use the publicly available dataset released by Yelp dataset¹. The dataset contains user-item interactions in the form of ratings and textual user reviews along with business features. The dataset contains 144,072 businesses, located in United States of America (USA), United Kingdom, Canada and Germany. These business are categorized into 1191 categories such as restaurants, nightlife, religious organizations, etc. Out of 45,472 users, we consider only users who have reviewed at least five businesses globally which reduces the number of users to 11609 and number of businesses to 1983. With no loss of generality, we consider the reviews of businesses tagged as “restaurants”, located in Phoenix city, USA. The basic dataset description is provided in Table 1.

5 Experimental Findings

In this section, we discuss the experiments conducted to evaluate our proposed topic modeling algorithm (*eLDA*) against benchmark models.

¹ https://www.yelp.com/dataset_challenge.

Table 1. Dataset description

Number of users	11,609
Number of businesses	1,983
Number of reviews	51,236
Business category	Restaurants
Location	Phoenix, AZ, USA

5.1 Benchmark Models

We have chosen three unsupervised models *TF-IDF*, *standard LDA* and *local-LDA* as benchmark models for evaluating the performance of our proposed model.

1. Term Frequency-Inverse Document Frequency (TF-IDF) [28]: This algorithm weighs the term frequency (TF) weighted by the logarithm of the inverse of the proportion of documents containing that term. A document-term matrix is then generated, with each document containing the TF-IDF score for all the terms in the vocabulary. K-means clustering technique is then employed for clustering the documents.
2. Standard LDA [5]: This algorithm treats each review as an individual document. Each document is further modeled as a probabilistic mixture of topics which themselves are distributed over the terms present in the vocabulary. This is a soft clustering technique that yields a probabilistic distribution of topics to a document.
3. Local LDA [6]: This algorithm is similar to *standard LDA* but treats each sentence as a document.

All the benchmark models are applied on review text which has been processed for removal of stopwords, numbers and punctuation marks. The corpus is then stemmed and lemmatized to obtain root words.

5.2 Intermediate Models

We introduce two intermediate models, “*LDA Noun-Codeword*” and “*LDA Noun*” to analyze the effect of extracting nouns and codeword insertion steps in isolation. “*LDA Noun-Codeword*” applies LDA on noun-codewords (some nouns replaced with noun-codewords) and each document corresponds to one review. “*LDA Noun*” applies LDA on nouns extracted from the reviews. In forthcoming sections, we compare the performance of our proposed models, *User eLDA* and *Item eLDA*, including the intermediate models against benchmark models.

5.3 Metrics for Evaluation

For evaluating the quality of the derived topics, we consider metrics such as perplexity, silhouette coefficient, word intrusion score and topic labeling. Quantitative metrics such as perplexity and silhouette coefficient are used to validate

the model performance whereas word intrusion and topic labeling are qualitative measures used for evaluating the quality of the derived topics.

1. **Perplexity:** Perplexity measures the predictive performance of a model given a set of unobserved documents. Perplexity is a monotonically decreasing function of log likelihood of unseen documents which is defined as follows:

$$Perplexity(D_{test}) = \exp\left(-\frac{\sum_{d \in \mathbb{T}} \log p(w_d)}{\sum_{d \in \mathbb{T}} N_d}\right) \quad (12)$$

Where D_{test} is the test corpus containing documents belonging to test set \mathbb{T} and N_d is the number of words in document d . Lower values of perplexity imply better performance of the model. Since *TF-IDF* does not have a log likelihood component, perplexity cannot be used to validate *TF-IDF*.

2. **Silhouette Coefficient:** Silhouette Coefficient measures how closely a document is assigned to its own cluster compared to other clusters. For a given document i , the silhouette coefficient is given as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (13)$$

Where $a(i)$ is the average distance between document i to all other documents in the same cluster and $b(i)$ is the lowest average distance between document i to all other documents from other clusters. Silhouette coefficient can range from -1 to 1 , where 1 implies a document is well assigned to its own cluster than other clusters and -1 implies the opposite.

3. **Word Intrusion:** Word intrusion task aids in quantitatively measuring the coherence of the topics [8]. In word intrusion task, a subject is presented with a set of most probable terms from a topic along with a randomly selected intruder term that does not belong to that topic. The task of the subject is to find the intruder term. For example, for a given set of words like $\{summer, winter, spring, autumn, dog\}$, a subject can easily identify *dog* as the intruder since all other words refer to seasons. When the terms in a topic lack such coherence, it becomes very difficult to identify the intruder term and subjects may randomly choose the intruder. Model precision is defined as follows:

$$MP_k^m = \frac{1}{S} \sum_s \mathbb{1}(i_{t,s}^m = w_k^m) \quad (14)$$

where MP_k^m is the word intrusion score generated for topic k and inferred from model m . Also, $i_{t,s}^m$ is the intruder word chosen by subject s and w_k^m is the actual intruder word. $\mathbb{1}(\cdot)$ is equal to 1 if the intruder word identified by the subject is same as actual intruder word w_k^m and 0 otherwise. Final word intrusion score for a model is the average word intrusion score for all the topics.

4. **Topic Labeling:** Topic labeling is the process of tagging domain-specific names to each of the topics. The degree to which subjects agree with the labeled terms for the topics can be quantified using Kappa statistics [31]. Cohen's Kappa is defined as follows.

$$\kappa = \frac{P_0 - P_c}{1 - P_c} \quad (15)$$

where P_0 is the proportion of mutual agreement and P_c is the proportion of agreement by chance. The values for Kappa can range from -1 to 1 , where 1 represents perfect agreement, -1 perfect disagreement and 0 agreement by chance.

5.4 Model Performance

Predictive Power. In this section, we evaluate the performance of intermediate models against benchmark models. To validate the impact of using noun codewords to extract topics, we first compare the performance of intermediate models *LDA Noun* and *LDA Noun-Codeword* with benchmark models. Table 2 summarizes the perplexities for the held out documents for number of topics (K) varied from 5 to 100. Note that *TF-IDF* is not reported in Table 2 as it does not have a log likelihood component to measure perplexity.

Table 2. Performance comparison with model perplexity

Number of topics	Local LDA	Standard LDA	LDA noun	LDA Noun-Codeword
5	1013.11	1014.84	784.74	231.94
15	867.81	907.44	674.93	181.76
30	744.19	823.99	587.97	162.21
50	649.85	758.81	528.28	150.15
75	582.03	709.64	480.20	142.54
100	539.52	674.55	453.36	137.22

As we can see from Table 2, *LDA Noun-Codeword* significantly outperforms the benchmark models. This can be attributed to two reasons: firstly, by replacing most of the food terms in the nouns with codewords reduces the number of terms in the vocabulary. Secondly, it reduces the confusion of assigning codewords to specific cuisine topics. For example, “pizza” being a very popular food term can belong to multiple topics like “American”, “Continental”, etc. and assigning “pizza” to the relevant topic for a given document becomes challenging. However, information about the cuisines offered by each restaurant can be used to replace these food terms with cuisine terms. For example “pizza” can be replaced with “American” if the restaurant serves American food. The task is now simplified to assign these codewords to cuisine topics and therefore identifying each entity’s cuisine preferences becomes easier. Also it helps to uncover other aspects of the restaurants like deals, hangout place etc. which remained hidden otherwise.

Cluster Quality. Table 3 summarizes the silhouette coefficients for all the models for varied number of topics. For LDA based models, each document is considered as one datapoint and the most probable topic as the cluster the document is assigned to. Each document’s topic probabilities are used as distance vectors for computing distance. For *TF-IDF*, the document-term matrix with TF-IDF scores is used for measuring distance. We use Euclidean distance to compute the distance between the documents for all the models.

Table 3. Performance comparison with silhouette coefficient

Number of topics	TF-IDF	Local LDA	Standard LDA	LDA Noun-Codeword
5	0.016	0.281	0.234	0.312
15	0.005	0.290	0.114	0.111
30	-0.015	0.118	0.069	0.07
50	-0.013	0.092	0.051	0.046
75	-0.01	0.075	0.038	0.036
100	-0.029	0.067	0.029	0.031

As seen from Table 3, *Local LDA* has highest silhouette coefficient compared to other models. One should note that silhouette coefficient assumes hard clustering, where one document can be assigned to only one cluster. The silhouette coefficient for *Local LDA* is relatively high because *Local LDA* treats each sentence as one document and the number of topics discussed in a sentence is much lower than number of topics discussed in an entire review. Therefore, measurement with silhouette coefficient alone will not be a determinant factor for model performance. To compare the quality of the topics, we will use other metrics as well.

Performance of Entity Level Models. To determine the efficacy of the proposed approaches (*User eLDA*, *Item eLDA* and *LDA Noun-Codeword*), perplexity and silhouette coefficient are measured for different number of topics.

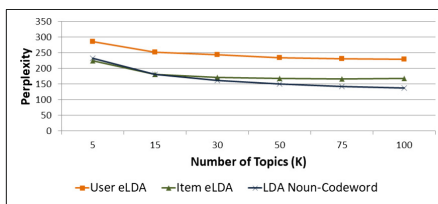


Fig. 2. Performance comparison with Model Perplexity

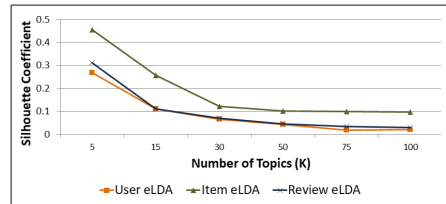


Fig. 3. Performance comparison with Silhouette Coefficient

Figure 2 demonstrates the superiority of *LDA Noun-Codeword*, for higher values of K with lowest perplexity. However, from Fig. 3, it can be observed that the silhouette coefficients for *Item eLDA* performs better compared to other models. Figure 4 compares the performance of Noun-Codeword models for 50 topics with respect to both perplexity and silhouette coefficient. Model performance is said to be better if the model perplexity is low and silhouette coefficient is high. To test the statistical significance of the performance of different models in terms of perplexity and silhouette coefficient, we perform a one-tailed two sample t-test for different number of topics. Table 4 summarizes the results of the statistical test.

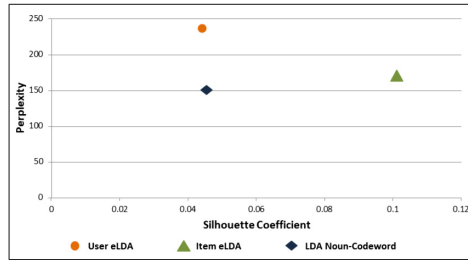


Fig. 4. Performance comparison of Noun-Codeword models (k = 50)

Table 4. Performance comparison with statistical significance

Metrics	User eLDA			Item eLDA			LDA Noun-Codeword		
	Mean	Std. dev.	p-value	Mean	Std. dev.	p-value	Mean	Std. dev.	p-value
Perplexity	246.43	21.31	0.0007***	179.97	22.58	0.25	167.64	35.29	-
Silhouette coefficient	0.088	0.095	0.41	0.189	0.144	0.87	0.101	0.108	-

Significance codes: $p < 0.01$ *** $p < 0.05$ ** $p < 0.1$ *.

The statistical test on *User eLDA* perplexity yields a p-value of 0.0007, which signifies the mean perplexity for *User eLDA* is statistically higher than *LDA Noun-Codeword* perplexity. The statistical test on *Item eLDA* perplexity yields a p-value of 0.25, and implies that mean perplexity for *Item eLDA* is statistically lower or not different from *LDA Noun-Codeword* perplexity. Although the perplexity of *User eLDA* is relatively higher than *LDA Noun-Codeword*, the performance of *User eLDA* is exceptionally exceeding compared to the traditional models (65.23% and 69.46% reduction in perplexity compared to *Local LDA* and *Standard LDA* respectively).

The statistical tests on silhouette coefficient yield p-values of 0.41 and 0.87 for *User eLDA* and *Item eLDA* respectively and therefore signify the mean silhouette coefficient for *User eLDA* and *Item eLDA* are statistically higher or not different from *LDA Noun-Codeword* silhouette coefficient.

The benefits of aggregating the noun-codewords at entity level (user/item) is multifaceted. Firstly, the size of the document-term matrix reduces drastically through aggregation and large number of terms can be accommodated in the document-term matrix. The number of documents in document-term matrix in this case have reduced by 77.34% and 96.12% for *User eLDA* and *Item eLDA* respectively, compared to *LDA Noun-Codeword*. Since the size of the document-term matrix is reduced, a reduction in the computational time can also be expected. Lastly, profiling the entities becomes much simpler as each document corresponds to an individual entity and the topic probabilities for each document can simply be used to represent entities' preference vectors.

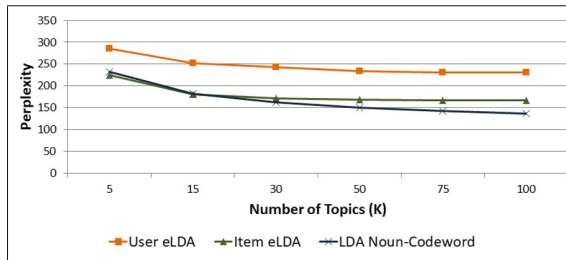


Fig. 5. Perplexity as a function of number of topics

Selecting the Number of Topics (K). To choose the optimal number of topics (K), held out perplexity is plotted as a function of number of topics for Noun-Codeword models. As seen from Fig. 5, the perplexity for the models is a decreasing function of number of topics, but remains stable after 50. Therefore, we pick 50 as the optimal topic number for our dataset.

5.5 Quality of Topics

The metrics discussed in Sect. 5.4 are crucial for evaluating model's performance. However, it is equally important to validate the quality of the derived topics. Herein, we present word clouds for each of the models, where the font size corresponds to the probability of the terms occurring in the topics. Later, we validate the quality of the topics using word intrusion and topic labeling technique.

Word Clouds. Due to space constraint, five random topics are selected from each of the models and are presented using word clouds. The size of each terms corresponds to the term probability for a given topic. The five topics include the following: cuisine (2), location (1), customer service (1) and order delivery (1). As seen from Fig. 6, for Noun-Codeword models, i.e *LDA Noun-Codeword*, *User eLDA* and *Item eLDA*, the term probabilities of noun codewords in cuisine topics ("Continental", "American") are exceptionally high. Similarly, the key terms for other topics such as location, customer service and order delivery have

a relatively higher probabilities compared to other models. This visual representation depicts the relevance of terms in a topic for Noun-Codeword models, as the term probabilities are relatively higher compared to other models.

Topic Labeling. Topic labeling is a crucial step for understanding the context of the derived topics. Manually labeling topics ensures high labeling quality [8]. We selected top 15 topics (based on mean topic probabilities) from each model and employed a random user to manually label the topics. Two subjects were asked to rate these topic labels as “relevant” or “irrelevant”. Cohen’s Kappa is used to measure the inter-rater agreement for the labeled topics. Table 5 shows the Kappa statistics for different models.

Table 5. Cohen’s Kappa for topic labeling

User eLDA	Item eLDA	LDA Noun-Codeword	Standard LDA	Local LDA	TF-IDF
0.42	0.66	0.63	0.19	0.24	0.21

It can be observed from Table 5 that Kappa statistic is high for Noun-Codeword models (*User eLDA*, *Item eLDA*, *LDA Noun-Codeword*). This indicates that our proposed model is able to generate of high quality topics and thereby reduces the chances of mislabeling the topics.

Word Intrusion. For evaluating the coherence of terms in topics, word intrusion task is performed. Top five terms based on high term probabilities are chosen from five topics, from all the models (shown in Fig. 6). For each of the topics, a low probable term from the same topic is chosen as intruder word. These set of terms are presented to 15 subjects. The task of the subjects is to find the intruder word from the given set of words. Since the probability of cuisine specific term in cuisine topics for Noun-Codeword models is extremely high, the cuisine term in isolation is sufficient for representing the entire topic. For example: in Fig. 6, *User eLDA* Topic 4 depicts that term “*Continental*” has a very high probability and is a representative of the entire topic. This eliminates the need for performing word intrusion task for cuisine topics for Noun-Codeword models and we argue that the model precision is always 1 for cuisine topics. For other benchmark models, we perform word intrusion on five topics whereas for Noun-Codeword models (*User eLDA*, *Item eLDA* and *Review Noun-Codeword*) we perform word intrusion task on only three non-cuisine topics (location, customer service and order delivery). Figure 7 presents model precision for all the models.

Figure 7 shows that Model Precision is high for Noun-Codeword models (*User eLDA*, *Item eLDA* and *Review Noun-Codeword*), which demonstrates better semantic coherence in the inferred topics.

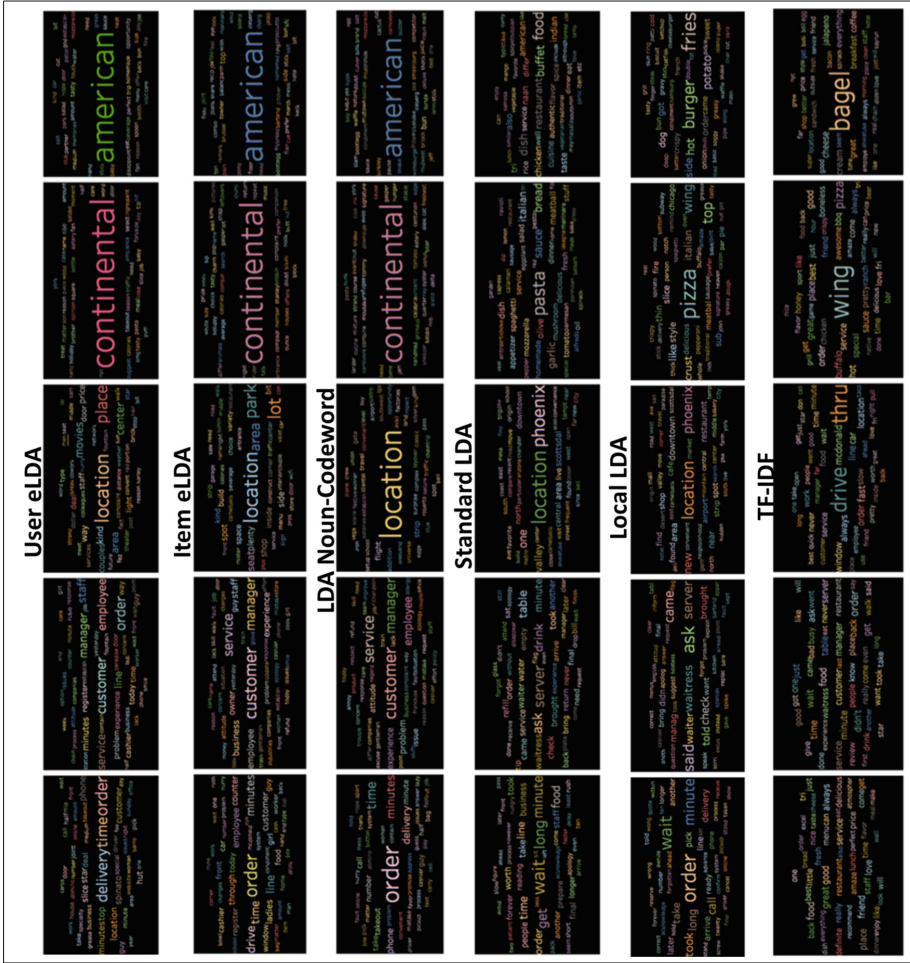


Fig. 6. Word cloud

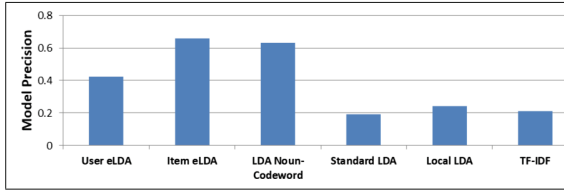


Fig. 7. Performance comparison with model precision

5.6 Sentiment Analysis

Sentiment Classifier Selection. We implement supervised machine learning sentiment classification algorithms to classify the sentiments associated with each of the aspects as positive or negative. We evaluate supervised classification algorithms commonly cited in sentiment analysis literature including logistic regression, K-Nearest neighbours (KNN), Gradient Boosting classifier, Random Forest classifier and Naive Bayes Classifier [24, 25, 29] to test their efficacy in classifying restaurant reviews. We use a dataset containing 1000 unique Yelp restaurant reviews² with sentiment labels for training our classifiers. The standard precision, recall, F-measure, and accuracy of the models, as obtained after an average of five random 80%–20% train-test splits are shown in Table 6.

Table 6. Sentiment classifier model performance

Model	Precision	Recall	F-measure	Accuracy
Gradient Boosting	0.752	0.707	0.729	0.737
Logistic Regression	0.806	0.757	0.780	0.793
Naive Bayes	0.656	0.856	0.742	0.700
Random Forests	0.863	0.620	0.721	0.755
KNN	0.689	0.675	0.680	0.687

Clearly, logistic regression outperforms the other algorithms in terms of F-measure score and overall accuracy. Therefore we select logistic regression as the sentiment classifier (\mathbb{K}) because of its high accuracy and well-balanced performance in terms of precision and recall.

6 Applications of Entity Profiles

6.1 Example of Profile Generation

The user and business profiles generated using our methodology have a multitude of potential applications. Our methodology generates the users and businesses

² <https://github.com/Microsoft/microsoft-r/tree/master/microsoft-ml/Samples/101/BinaryClassification/SimpleSentimentAnalysis>.

profiles using review texts, thus reflecting the popular preferences and perspectives about the businesses. We demonstrate one example each for profiling users and businesses from a sample review taken from the Yelp dataset.

An Example of User Profiling. Consider a snippet of the user-level document for one user: *‘Decent food, decent beer, fast service what more do you want? Price was fair too ! Some TVs to catch sports. Nothing special but gets the job done. Okay food, drinks, waitresses etc. The highlight is the karaoke...’*

We can intuitively say that this user is interested in food, drinks, service, prices and overall ambience based on the fact that he/she has mentioned these attributes. We demonstrate step-wise generation of the profile for this particular user:

1. After the pre-processing stage, the nouns and codewords are extracted. Thus, the above text is transformed into: “AMERICAN ALCOHOL service price tvs sports nothing job AMERICAN drinks waitresses highlight karaoke....”. Thus, the text now contains only nouns and noun-phrase with the appropriate codewords, which are “AMERICAN” and “ALCOHOL” in this case, to denote the corresponding food category.
2. In the topic modelling stage, the pre-processed text is run through the *User eLDA* model to extract the corresponding topics T_u and the K -dimensional probability distribution vector p_u .
3. In the profile generation stage, a manually generated mapping \mathcal{M} is used to map the $K = 50$ topics in T_u to the k user profile categories. In our case, we have $k = 15$ categories, corresponding to $\Pi = \{\text{american, continental, asian, indo-arabic, desserts, sandwiches, alcohol, nightlife, breakfast-brunch, southern, ambience, price, restaurant quality, service efficiency, location}\}$. Thus, the mapping \mathcal{M} would map the topics “good waiter”, “bad service”, “bad waiter” to the category “Service quality”. Also, the k -dimensional category probability vector P_u will be the aggregate of the K -dimensional topic probability vector p_u by taking the sum.
4. The user profile U_u is generated as: [0.147, 0, 0, 0, 0, 0, 0.031, 0, 0, 0.039, 0, 0, 0.031, 0.112, 0.147], where the $U_u[i]$ corresponds to Π_i for all $k \in [1, 15]$. The topic probabilities for most of the terms in this example are very low and hence was rounded to zero. We can directly infer that the user has a very high interest in American cuisine (0.147), restaurant locations (0.147) and service efficiency (0.112), while being mindful of the restaurant quality (0.031), nightlife (0.031), and southern cuisine (0.039). This finding corroborates well with our initial intuition about the user.

An Example of Item Profiling. Consider a snippet of the item-level document for one restaurant: *‘Brand new food business! They make you a pizza, you take it home and bake it. It’s a create your own pizza place... But terrible service...’*

We can infer that this restaurant serves a take-at-home pizza, which is satisfactory in terms of food, but poor in service. We can generate the item profile I_i for this restaurant following the steps similar to those mentioned for generating user profiles. This item’s profile obtained from *Item eLDA* model is [0.34, 0.04, 0, 0, 0, 0, 0.03, 0, 0, 0.04, 0, 0, 0.031, 0.23, 0]. Further, we generate the aspect level sentiment-polarities as follows:

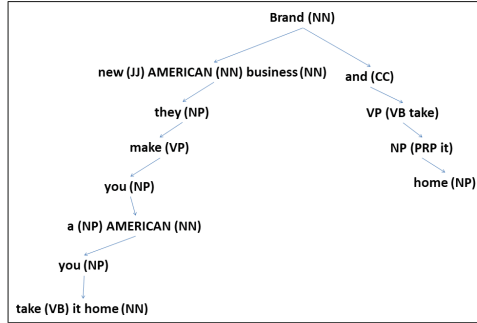


Fig. 8. Example of a parse tree for a single sentence

1. Firstly, we generate the parse tree from the text preprocessing stage as depicted in Fig. 8.
2. We then derive the aspect level sentiment score as explained in Sect. 3.3. Thus we obtain the following aspects and corresponding scores: {new american business: 0.83, AMERICAN: 0.69, home: 0.41 own pizza place: 0.63 and service: 0.14}.
3. We then aggregate the aspect level sentiments to category level sentiments using arithmetic mean: In this case, the sentiment for ‘American’ category would be an aggregate of sentiment scores of ‘American’, ‘new american business’ and ‘own pizza place’, which is 0.72. The aggregated sentiment scores for home and service correspond to service efficiency category and will be 0.28.
4. Then the sentiment polarities are calculated as follows: American having a score of 0.59 will take a positive score of +2 and service with a score of 0.28 will take a negative sentiment score of -1.
5. The topic probabilities for this business obtained from *Item eLDA* is [0.34, 0.04, 0, 0, 0, 0, 0.03, 0, 0, 0.04, 0, 0, 0.031, 0.23, 0]. Now the category sentiment polarities will be multiplied to the topic probabilities to generate the business profile I_i as: $I_i = [0.68, 0.04, 0, 0, 0, 0, 0.03, 0, 0, 0.04, 0, 0, 0.031, -0.23, 0]$. This vector will represent the business profile for the specified business.

We can infer from the generated business profile that the restaurant is perceived to have a very good american cuisine, a decent continental cuisine, alcohol

service, southern cuisines and overall quality, but bad service efficiency. Thus, the generated business profile closely reflects the inferred intuition about the restaurant.

6.2 Proposed Applications

The applications of these personalized profiles are multifaceted. We envision their direct applications in the following fields:

Recommender Systems. Recommender systems generate personalized recommendations by clustering similar users and items based on historical ratings [20,35]. Similarity measurements can be directly refined by incorporating the user and business profiles generated from text reviews. Further, our profiles can explain the recommendations in a more natural manner, by weighing in the user's interests and the business's strengths and weaknesses.

Web Personalization. Web personalization customize online content to suit individual users and effectively project the businesses [10]. Generated profiles can potentially make a twofold impact in this sector. Personalization of online content based on the generated profiles can potentially improve click-through and conversion rates.

Marketing. Traditionally, users and businesses are profiled using historical rating and purchase history to classify them into various market segments [30]. However, profiles generated using online reviews have the potential to unravel attributes such as price, quality, ambiance etc. which in-turn can improve the existing customer segmentation process.

7 Conclusion

In this paper, we propose an entity-level topic model - *eLDA*, which incorporates domain-specific noun-codewords aggregated at entity-levels (user and item levels) to derive underlying abstract topics hidden in textual reviews. Several experiments have been conducted on a large review dataset of restaurants to validate the predictive performance and quality of derived topics. Experimental results reveal that our proposed models, *User eLDA* and *item eLDA*, outperform other benchmark models with improved topic coherence. The topic probabilities obtained from *eLDA* models are then mapped to domain-specific aspects for building entity profiles. This pivotal step for personalization reveal the actual meaning of these latent features and hence improves the ease of interpretability, which was largely overlooked by the traditional CF-based personalized algorithms. The aggregation of documents to the entity levels reduces the size of large Document-Term Matrix (DTM) drastically which ensures a huge reduction in the computational complexity as well. Findings show that there is a gratifying

reduction of 77.34% and 96.12% in DTM size for *User eLDA* and *item eLDA*, respectively compared to *LDA Noun-codeword*, with good predictive power and topic quality. In our future work, we plan to fuse the entity profiles to an existing recommender system algorithm and validate the quality of the recommendations.

References

1. Bao, Y., Datta, A.: Simultaneously discovering and quantifying risk types from textual risk disclosures. *Manag. Sci.* **60**(6), 1371–1391 (2014)
2. Bao, Y., Fang, H., Zhang, J.: TopicMF: simultaneously exploiting ratings and reviews for recommendation. In: *AAAI*, vol. 14, pp. 2–8 (2014)
3. Bird, S., Klein, E., Loper, E.: *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., Sebastopol (2009)
4. Blei, D.M., Lafferty, J.D.: Correlated topic models. In: *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pp. 147–154. MIT Press (2005)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**(Jan), 993–1022 (2003)
6. Brody, S., Elhadad, N.: An unsupervised aspect-sentiment model for online reviews. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 804–812. Association for Computational Linguistics (2010)
7. Cao, X., Cong, G., Jensen, C.S.: Mining significant semantic locations from GPS data. *Proc. VLDB Endow.* **3**(1–2), 1009–1020 (2010)
8. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: how humans interpret topic models. In: *Advances in Neural Information Processing Systems*, pp. 288–296 (2009)
9. Cheng, C., Yang, H., King, I., Lyu, M.R.: Fused matrix factorization with geographical and social influence in location-based social networks. In: *AAAI*, vol. 12, pp. 17–23 (2012)
10. Eirinaki, M., Vazirgiannis, M.: Web mining for web personalization. *ACM Trans. Internet Technol. (TOIT)* **3**(1), 1–27 (2003)
11. Feng, H., Qian, X.: Recommendation via user's personality and social contextual. In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 1521–1524. ACM (2013)
12. Hofmann, T.: Probabilistic latent semantic analysis. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 289–296. Morgan Kaufmann Publishers Inc. (1999)
13. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: *AAAI*, vol. 4, pp. 755–760 (2004)
14. Jakob, N., Gurevych, I.: 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*, pp. 1035–1045. Association for Computational Linguistics (2010)
15. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: *Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 135–142. ACM (2010)
16. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: GroupLens: applying collaborative filtering to usenet news. *Commun. ACM* **40**(3), 77–87 (1997)

17. Linshi, J.: Personalizing yelp star ratings: a semantic topic modeling approach. Yale University (2014)
18. Long, C., Zhang, J., Zhut, X.: A review selection approach for accurate feature rating estimation. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 766–774. Association for Computational Linguistics (2010)
19. Lu, B., Ott, M., Cardie, C., Tsou, B.K.: Multi-aspect sentiment analysis with topic models. In: 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW), pp.81–88. IEEE (2011)
20. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 287–296. ACM (2011)
21. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford coreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association For Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
22. Martin, F., Johnson, M.: More efficient topic modelling through a noun only approach. In: 2015 Proceedings of the Australasian Language Technology Association Workshop, pp. 111–115 (2015)
23. Brusilovsky, P., Maybury, M.T.: From adaptive hypermedia to the adaptive web. *Commun. ACM* **45**(5), 30–33 (2002)
24. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. *Ain Shams Eng. J.* **5**(4), 1093–1113 (2014)
25. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
26. Qiu, G., Liu, B., Bu, J., Chen, C.: Expanding domain sentiment lexicon through double propagation. In: IJCAI, vol. 9, pp. 1199–1204 (2009)
27. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM (1994)
28. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **24**(5), 513–523 (1988)
29. Schouten, K., Frasincar, F.: Survey on aspect-level sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **28**(3), 813–830 (2016)
30. Shaw, M.J., Subramaniam, C., Tan, G.W., Welge, M.E.: Knowledge management and data mining for marketing. *Decis. Support Syst.* **31**(1), 127–137 (2001)
31. Sim, J., Wright, C.C.: The Kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Phys. Ther.* **85**(3), 257–268 (2005)
32. Socher, R., Bauer, J., Manning, C.D., et al.: Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Long Papers, vol. 1, pp. 455–465 (2013)
33. Titov, I., McDonald, R.: A joint model of text and aspect ratings for sentiment summarization. In: Proceedings of ACL 2008: HLT, pp. 308–316 (2008)
34. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 783–792. ACM (2010)

35. Yang, D., Zhang, D., Yu, Z., Wang, Z.: A sentiment-enhanced personalized location recommendation system. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media, pp. 119–128. ACM (2013)
36. Ye, M., Yin, P., Lee, W.C., Lee, D.L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325–334. ACM (2011)
37. Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 56–65. Association for Computational Linguistics (2010)
38. Zhao, Y., Qin, B., Hu, S., Liu, T.: Generalizing syntactic structures for product attribute candidate extraction. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 377–380. Association for Computational Linguistics (2010)
39. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with GPS history data. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1029–1038. ACM (2010)
40. Zheng, V.W., Cao, B., Zheng, Y., Xie, X., Yang, Q.: Collaborative filtering meets mobile recommendation: a user-centered approach. In: AAAI, vol. 10, pp. 236–241 (2010)
41. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web, pp. 791–800. ACM (2009)