Using Neural Word Embeddings to Model User Behavior and Detect User Segments

Ludovico Boratto, Salvatore Carta, Gianni Fenu, and Roberto Saia[1]

Dipartimento di Matematica e Informatica, Università di Cagliari
Via Ospedale 72, 09124 Cagliari, Italy

Abstract

Modeling user behavior to detect segments of users to target and to whom address ads (*behavioral targeting*) is a problem widely-studied in the literature. Various sources of data are mined and modeled in order to detect these segments, such as the queries issued by the users. In this paper we first show the need for a user segmentation system to employ *reliable* user preferences, since nearly half of the times users reformulate their queries in order to satisfy their information need. Then we propose a method that analyzes the description of the items positively evaluated by the users and extracts a vector representation of the words in these descriptions (word embeddings). Since it is widely-known that users tend to choose items of the same categories, our approach is designed to avoid the so-called *preference stability*, which would associate the users to trivial segments. Moreover, we make sure that the *interpretability* of the generated segments is a characteristic offered to the advertisers who will use them. We performed different sets of experiments on a large real-world dataset, which validated our approach and showed its capability to produce effective segments.

Keywords: User Segmentation, Semantic Analysis, Behavioral Targeting, Word Embeddings

1. Introduction

Behavioral targeting is the process of detecting segments of users with similar behaviors, in order to address effective ads to them. Given the high interest that extracting effective segments has, both the industry and the academia are studying ways to model user behavior. In the industry, the systems try to monitor the behavior of the users in implicit ways, in order to extract their preferences and form the segments; usually, neither the algorithms not the data are publicly

Email address: {ludovico.boratto,salvatore,fenu,roberto.saia}@unica.it (Ludovico Boratto, Salvatore Carta, Gianni Fenu, and Roberto Saia)

made available, to avoid disclosing both industrial secrets and private information about the users. In the academic literature it has been highlighted that classic approaches to segmentation (like k-means) cannot take into account the semantics of the user behavior [1]. Tu and Lu [2] proposed a user segmentation approach based on a semantic analysis of the queries issued by the users, while Gong et al. [1] proposed a LDA-based semantic segmentation that groups users with similar query and click behaviors.

However, several problems remain open in the literature when performing a user segmentation by considering the user behavior.

Data sources reliability. In order to satisfy the users' information need, query reformulation characterizes nearly 50% of the queries issued by the users [3, 4, 5]. Therefore, the semantic analysis of a query is not a reliable source of information, since it does not contain any information about whether or not a query led to what the user was really looking for. Moreover, performing a semantic analysis on the items evaluated by the users, in order to perform a filtering on them, can increase the accuracy of a system [6, 7, 8]. Considering these aspects, a possible solution to this issue would be a semantic analysis on the description of the items a user positively evaluated through an explicitly given rating. However, another issue arises in cascade.

Preference stability. The analysis of reliable information about the users, such as the description of the items evaluated by them, would probably lead to trivial segments, since users tend to evaluate items of the same categories (e.g., they usually watch movies of the same genres or by the same director/actor). This problem is known as *preference stability* [9] and on the one hand it leads to high-quality knowledge sources, while on the other hand there is no way to target the users with serendipitous and effective ads (*overspecialization* [10]).

Segmentation interpretability. Another open issue widely-studied in this research area is the capability for a segmentation to be easily interpreted. A recent survey on user segmentation (mostly focused on the library domain) [11], highlighted that, in order to create a proper segmentation of the users, it is important to *understand* them. On the one hand, easily interpretable approaches generate trivial segments, and even a partitioning with the k-means clustering algorithm has proven to be more effective than this method [12], while on the other hand, when a larger set of features is combined, the problem of properly understanding and interpreting results arises [13, 14]. This is mostly due to the lack of guidance on how to interpret the results of a segmentation [15]. The fact that easily understandable approaches generate ineffective segments, and that more complex ones are accurate but not easy to use in practice, generates an important gap in this research area.

Our contributions. In this paper, we present an approach to user segmentation, such that the sources of information used to build it are reliable, the generated user segmentation is not trivial and it is easily interpretable.

As previously mentioned, the problem of using reliable sources of information will be solved by considering the items positively evaluated by the users with an explicitly-assigned rating. In particular, we employ the vector representation of the words in a description (word embedding) [16]. Word embeddings

are built by considering as input a text corpus, that leads to the building of a vocabulary, and to the learning of the vector representation of the words. They are largely employed nowadays in several NLP tasks, such as the representations of sentences and paragraphs [17, 18], relational entities [19, 20], general text-based attributes [21], descriptive text of images [22], and nodes in graph structure [23]. According to the authors' knowledge, no approach uses word embeddings for user segmentation purposes.

For each class of items that the users can be targeted with (e.g., movie genres), our approach builds a vector representation based on the word embeddings, which characterizes the words that represent the class. In a similar way, we also build a user model that captures the user interests. By matching the vector representations of a class of items with the user model, thanks to a similarity metric, we can associate a user to the segment that represents that class of items. It is trivial to notice that each user can have a strong similarity also with classes of items she never evaluated (thus avoiding the preference stability problem) and that the segments can be easily interpreted (even if a class and a user are represented by tens of features in the vector, the advertiser is only required to specify which interests she wants to target). In order to allow advertisers to specify more complex targets, we also present a boolean algebra that combines multiple classes of items with simple operations (e.g., to extract a vector representation of what characterizes *comedy AND romantic* movies).

More formally, the problem statement is the following:

Problem 1. *We are given a set of users* $U = \{u_1, \ldots, u_N\}$*, a set of items* $I = \{i_1, \ldots, i_M\}$*, and a set* R *of ratings used to express the user preferences (e.g.,* $R = [1, 5]$ *or* $R = \{like, dislike\}$*). The set of all possible preferences expressed by the users is a ternary relation* $P \subseteq U \times I \times R$*. We denote as* $P_+ \subseteq P$ *the subset of preferences with a positive value, as* I_+ *the items for which there is a positive preference, and as* I_u *the items positively evaluated by a user* u*. The set of item descriptions is denoted as* $D = \{d_1, \ldots, d_M\}$ *(note that we have a description for each item, so* $\mid D \mid = \mid I \mid$*), and the vocabulary of the words in* D *is denoted as* $V = \{v_1, \cdots, v_W\}$*. Let* $NWE_{v_w} = \{l_1, \cdots, l_Z\}$ *be the vector representation (*neural word embedding*) of each word* $v_w \in V$*. We denote as* $C = \{c_1, \ldots, c_K\}$ *the set of* primitive classes *used to classify the items. Our first aim is to extract a vector representation of each class* $c_k \in C$ *based on the neural word embeddings of the description of the items classified with* c_k *(*neural class embedding*,* NCE*), and a vector representation of each user* $u \in U$ *(*user model*,* m_u*). The objective of this paper is to build a function* $f : C \rightarrow U$ *that, given a class, returns a set of users to target* $T \subseteq U$*, such that the similarity between the neural class embedding and the models of the users in* T *is higher than a threshold value.*

The scientific contributions of our proposal are now presented:

- we propose a novel use of neural word embeddings for user segmentation purposes;

- we introduce a novel data structure, called *NCE* (*Neural Class Embedding*), able to model the words that characterize a class of items;
- we consider, for the first time in the user segmentation literature, the reliability of the data sources. Indeed, with respect to the literature that usually performs an analysis of the queries issued by the users, we rely on the description of the items a user positively rated;
- we avoid preference stability by considering the similarity between each user model and the vector representation of a classes of items, in order to allow the approach to include a user in a segment that represents a class of items she has never evaluated, but that is highly similar to her preferences;
- we present a boolean algebra that allows us to specify, in a simple but punctual way, the interests that the segment should cover; the algebra, along with the built models, avoids the interpretability issues that usually characterize the segmentations based on several features.

The rest of the paper is organized as follows: we first present the works in the literature related with our approach (Section 2), then we continue with the implementation details (Section 3) and the description of the performed experiments (Section 4), ending with some concluding remarks (Section 5).

2. Related Work

Here, we report the main approaches developed in the industry and in the literature for each of the topics related to our work.

Behavioral targeting. Most of the approaches to behavioral targeting have been developed by the industry as real-word systems. Among the different types of targeting that Google's *AdWords*[1] developed to present ads to the users, the closest to our proposal is "Topic targeting", in which the system groups and reaches the users interested in a specific topic. In order to detect segments that contain similar users, Facebook offers *Core Audiences*[2], a tool that allows advertisers to target users with similar locations, demographics, interests, or behaviors; in particular, the interest-based segmentation considers a topic and targets a segment of users interested by it. The service offered by Amazon, instead, is called *Interest-based ads policy*[3], and it targets segments of users with similar interests, based on what the users purchased and visited, and by monitoring different forms of interaction with the website (e.g., the Amazon Browser Bar). *Yahoo! Behavioral Targeting*[4] creates a model with the online interactions of the users, such as searches, page-views, and ad interactions, to predict

[1] https://support.google.com/adwords/answer/1704368?hl=en
[2] https://www.facebook.com/business/news/Core-Audiences
[3] http://www.amazon.com/b?node=5160028011
[4] http://advertising.stltoday.com/content/behavioral_FAQ.pdf

the set of users to target. Other commercial systems, such as *DoubleClick*[5], *SpecificMedia*[6], *Almond Net*[7], *Burst*[8], *Phorm*[9], and *Revenue Science*[10] include behavioral targeting features. When considering the studies presented in the literature, there are approaches that exploit the semantics [6, 7] or the capabilities of a recommender system [24, 25, 26] to improve the effectiveness of the advertising, but none of them generates segments of target users. Yan et al. [27] instead perform online advertising by monitoring the click-through log of advertisements collected from a commercial search engine. Beales [28] studied that prices and conversion rates (i.e., the likelihood of a click to lead to a sale), collected from online advertising networks, are more effective sources of data to decide which ads should be presented to the users. Chen et al. [29] presented a scalable approach to behavioral targeting, based on a linear Poisson regression model that uses granular events (such as individual ad clicks and search queries) as features.

Behavioral user segmentation and segment interpretability. Bian et al. [30] presented an approach to leverage historical user activity on real-world Web portal services to build a behavior-driven user segmentation. Yao et al. [31] adopted SOM-Ward clustering (i.e., Self Organizing Maps, combined with Ward clustering), to segment a set of customers based on their demographic and behavioral characteristic. Zhou et al. [32] performed a user segmentation based on a mixture of factor analyzers (MFA) that consider the navigational behavior of the user in a browsing session. Regarding the semantic approaches to user segmentation, Tu and Lu [2] and Gong et al. [1] both proposed approaches based on a semantic analysis of the queries issued by the users through Latent Dirichlet Allocation-based models, in which users with similar query and click behaviors are grouped together. Similarly, Wu et al. [33] performed a semantic user segmentation by adopting a Probabilistic Latent Semantic Approach on the user queries. As this analysis showed, none of the behavioral targeting approaches exploits the interactions of the users with a website in the form of a positive rating given to an item. Choosing the right criteria to segment users is a widely studied problem in the market segmentation literature, and two main classes of approaches exist. On the one hand, the *a priori* [34] or *commonsense* [35] approach is based on a simple property, like the age, which is used to segment the users. Even though the generated segments are very easy to understand and they can be created at a very low cost, the segmentation process is trivial and even a partitioning with the k-means clustering algorithm has proven to be more effective than this method [12]. On the other hand, *post hoc* [36] approaches (also known as *a posteriori* [34] or *data-driven* [35]) combine a set of features (which are known as *segmentation base* [37]) in order to create the segmentation.

[5] https://www.google.com/doubleclick/
[6] http://specificmedia.com/
[7] http://www.almondnet.com/
[8] http://www.burstmedia.com/
[9] http://www.phorm.com/
[10] http://www.revenuescience.com/

Even though these approaches are more accurate when partitioning the users, the problem of properly understanding and interpreting results arises [13, 14]. This is mostly due to the lack of guidance on how to interpret the results of a segmentation [15].

Preference stability. Most domains are characterized by a stability of the preferences over time [9]. Preference stability leads also to the fact that when users get in touch with diverse items, diversity is not valued [38]. On the one side, users tend to access to agreeable information (a phenomenon known as *filter bubble* [39]) and this leads to the overspecialization problem [10], while on the other side they do not want to face diversity. Another well-known problem is the so called *selective exposure*, i.e., the tendency of users to make their choices (goods or services) based only on their usual preferences, which excludes the possibility for the users to find new items that may be of interest to them [40]. The literature presents several approaches that try to reduce this problem, e.g., *NewsCube* [41] operates by offering to the users several points of view, in order to stimulate them to make different and unusual choices.

3. Using Word Embeddings to Model User Behavior and Detect User Segments

Here we present the algorithms employed to built our user segmentation. The approach works in five steps:

1. **Neural Word Embeddings extraction**: processing of the textual information of all the items, in order to remove the useless elements from the text (e.g., punctuation marks, stop words, etc.), and extract the word embeddings;
2. **Neural Item Embeddings extraction**: for each item, we sum the correspondent vector elements of the words that compose its description, to obtain as result a vector representation of the item;
3. **User Modeling**: creation of a model built by considering the embeddings of the items a user likes;
4. **Neural Class Embedding definition**: creation of the *Neural Class Embeddings* (*NCE*s), i.e., a series of vectors able to characterize each class of items, based on the word embeddings of the items that belong to a class; a class can either be a primitive class with whom an item was classified, or a *boolean* one that combines the primitive classes through boolean operators;
5. **User segmentation**: selection of the users characterized by a specified class, by comparing the user models with the NCE of the considered classes.

In the following, we will describe in detail how each step works.

3.1. Neural Word Embeddings Extraction

Before extracting the vector representation from each word that describes an item, we need to follow several preprocessing steps. Given an item description,

the first form of preprocessing is the conversion of the string into lowercase characters. Then we remove from the text punctuation marks, multiple spaces, and stop-words. The text, split into words, is processed by the framework to extract the neural word embeddings by using Google's *word2vec*[11], the most widely-employed implementation in the literature. The tool requires as input the item descriptions and a parameter Z that indicates how many layers (i.e., how many elements) the vector should have. The tool first constructs a vocabulary from the training text data and then learns the vector representation of the words. The output of this step is a set of vectors that we named NWE, which contain the representation of each word in the vocabulary as an embedding in Z layers. Each element of the vector contains the relevance of the layer for that word.

3.2. Neural Item Embeddings Extraction

For each item $i_m \in I$, we consider its description d_m and the embeddings of the words in it. The vector representation of the items is represented as the sum of the vector representation of the words in their descriptions. Since the vectors generated by the previous steps are linear, it is common to sum several vectors to represent a short sentence (additive compositionality property) [42]. In this study we refer to the vector representation of an item as *Neural Item Embedding* (NIE) and we build it as follows:

$$NIE_{i_m} = \begin{cases} NIE_{i_m} + NWE_{v_w},\ if\ v_w \in d_m \\ NIE_{i_m},\ otherwise \end{cases} \tag{1}$$

By performing this operation for all the words $v_w \in V$, we characterize each item by considering the relevance of each layer for that item, based on the neural word embeddings of its description.

3.3. User Modeling

For each user $u \in U$, this step considers the set of items I_u she likes, and builds a user model m_u that describes how each layer characterizes the user profile. Each model m_u is a vector that contains an element for each layer. In order to build the vector, we consider the NIE_{i_m} of each item $i_m \in I_u$ for which the user expressed a positive preference and perform the following operation:

$$m_u = \begin{cases} m_u + NIE_{i_m},\ if\ i_m \in I_u \\ m_u,\ otherwise \end{cases} \tag{2}$$

This means that if a user expressed a positive preference for item i_m, we add the relevance of each layer that represents the item to the user model m_u; otherwise, the value of the layer remains unaltered. By performing this operation for all the items $i_m \in I_u$, we model a user by considering the values of the layers in those items. The output of this step is a set $M = \{m_1, \ldots, m_N\}$ of user models (note that we have a model for each user, so $\mid M \mid = \mid U \mid$).

[11] https://code.google.com/archive/p/word2vec/

3.4. Neural Class Embedding Definition

Given a set of classes C, in this step we define a vector, called *Neural Class Embedding* (NCE), which represents each class as a vector, by considering the embeddings of the items evaluated with that class. Moreover, we are going to present an approach to build the *boolean classes* previously defined, i.e., a NCE that describes multiple classes combined through a set of boolean operators $\tau = \{\wedge, \vee, \neg\}$.

Therefore, four types of neural class embeddings can be defined:

1. ***Primitive class-based* NCE definition.** Given a primitive class of items c_k, this operation creates a vector that contains the relevance of each layer for that class, based on the embeddings of the items classified with c_k.
2. ***Interclass-based* NCE definition.** Given two classes c_k and c_q, we combine the NCEs of the two classes with an AND operator, in order to build a new neural class embedding that contains the layers that characterize both the classes.
3. ***Superclass-based* NCE definition.** Given two classes c_k and c_q, we combine the NCEs of the two classes with an OR operator, in order to build a new neural class embedding that merges the relevance of each layer for the two classes if it is characterizing for at least one of them.
4. ***Subclass-based* NCE definition.** Given two classes c_k and c_q, we use the NCE of c_q as a negation mask on the NCE of c_k, in order to build a new neural class embedding that contains the layers that characterize the first class but do not characterize the second.

3.4.1. Primitive class-based NCE Definition

For each class $c_k \in C$, we create a vector that stores the relevance of each layer for that class. These vectors, called *Neural Class Embedding*, will be stored in a set $NCE = \{nce_1, \ldots, nce_K\}$ (note that $\mid NCE \mid = \mid C \mid$, since we have a vector for each class). Each vector $nce_k \in NCE$ contains an element for each layer. In order to build the vector, we consider the NIE of each item $i_m \in I_+$ for which there is a positive preference, and each class c_k with whom i_m was classified. The vector nce_k will store the relevance of each layer for a class c_k, by considering the neural item embedding of the items classified with it:

$$nce_k = \begin{cases} nce_k + NIE_{i_m}, \ if \ i_m \in c_k \\ nce_k, \ otherwise \end{cases} \tag{3}$$

In other words, we sum the relevance of each layer of an item i_m to that of the class c_k that classifies i_m. By performing this operation for all the items $i_m \in I_+$ that are classified with c_k, we know the relevance of each layer for the class. After we processed all the classes in C, we obtain the description of the primitive classes that allow us to build the filters for the boolean classes.

3.4.2. Interclass-based NCE Definition

Starting from the set $NCE = \{nce_1, \ldots, nce_K\}$, we can arbitrarily manage the elements $nce_k \in NCE$ to generate *boolean classes*, i.e., the combination of primitive classes by means of a boolean operator. The first type of boolean class we are going to define, named *interclass* is formed by the combination of the neural class embeddings of the two classes nce_k and nce_q through an AND operator. Considering each element l_z of the two vectors, which indicates the relevance of a layer l_z for a class, the semantics of the operator is the following:

$$nce_k[l_z] \wedge nce_q[l_z] = \begin{cases} nce_k[l_z] + nce_q[l_z], \ if \ (nce_k[l_z] > 0) \ \wedge (nce_q[l_z] > 0) \\ 0, \ otherwise \end{cases} \tag{4}$$

This boolean class indicates which layers characterize all the involved classes of items. The relevance of a layer l_z for both classes is summed if both classes have have relevance higher than zero for that layer[12]. We can obtain this result recurring to the axiomatic set theory (i.e., the elementary set theory based on the Venn diagrams); indeed, we can consider each class of items as a set, and create a new interclass that characterizes the common elements of two or more NIEs, using an intersection operation $\cap$;

The example in Figure 1 is a simple demonstration of what said, based on the axiomatic set theory. It describes the effect of a boolean AND operation applied to the classes C_1, C_2, and C_3: in this case, the result of operation $C_1 \cap C_2 \cap C_3$ represents a new interclass that we can use to refer to a precise target of users, in a more atomic way than with the use of the primitive classes.

3.4.3. Superclass-based NCE Definition

By combining the neural class embeddings of two classes nce_k and nce_q through an OR operator, we can generate a new type of boolean class, named *superclass*. Considering each element l_z of the two vectors, which indicates the relevance of a layer l_z for a class, the semantics of the operator is the following:

$$nce_k[l_z] \vee nce_q[l_z] = \begin{cases} nce_k[l_z] + nce_q[l_z], \ if \ (nce_k[l_z] > 0) \ \vee (nce_q[l_z] > 0) \\ 0, \ otherwise \end{cases} \tag{5}$$

This boolean class would allow an advertiser to broaden a target, capturing in a neural class embedding the relevance of the layers that are characterizing for two or more classes. By using the axiomatic set theory, we can consider each class of items as a set, and create a new superclass that characterizes more native classes through an union operation $\cup$ of two or more SBSs.

[12]Note that we designed and developed several strategies and this is the most effective to understand when a layer is relevant for a class. We omit the other strategies and the associated experiments to facilitate the reading of the paper.

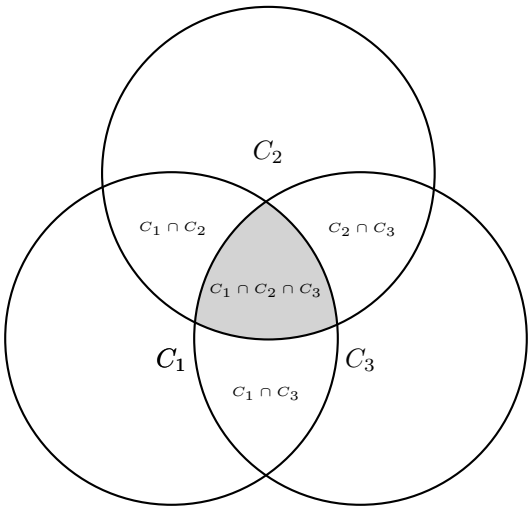

Figure 1: Inter-class definition

The example in Figure 2 shows demonstration of what said based on the axiomatic set theory. It describes the effect of a boolean *OR* operation applied to the classes C_1, C_2, and C_3 (represented by the grey area).

3.4.4. Subclass-based NCE Definition

Another important entity that we can obtain through the managing of the elements $nce_k \in NCE$ is the subset of a primitive class. It means that we can extract from a neural class embedding a subset of elements that express an atomic characteristic of the source set. For instance, if we consider a dataset where the items are movies, from a subset of native genres of classification we can extract neural class embeddings that characterize some sub-genres of movies.

More formally, a *subclass* is a partition of a primitive or boolean class, e.g., for the primitive class *Comedy* we can define an arbitrary number of subclasses, applying some operation of the axiomatic set theory. In the example in Figure 3, we define a subclass $Comedy \backslash Romance$, in which all the layers that characterize the *Romance* class are removed from the *Comedy* class. Therefore, only the comedy movies that do contain romance elements are represented through this boolean class.

Given two neural class embeddings nce_k and nce_q, each element l_z of the nce_k vector remains unaltered if the corresponding element of the nce_q is lower than or equal to zero, otherwise it is set to zero, as shown in Equation (6).

$$nce_k[l_z] = \begin{cases} nce_k[l_z], \; if \; nce_q[l_z] \leq 0 \\ 0, \; otherwise \end{cases} \tag{6}$$

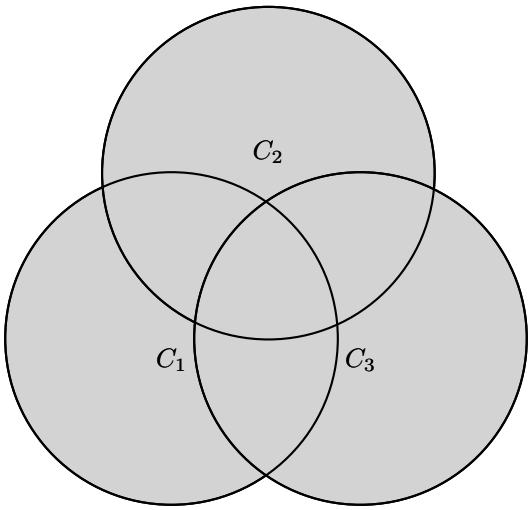

Figure 2: Superclass definition

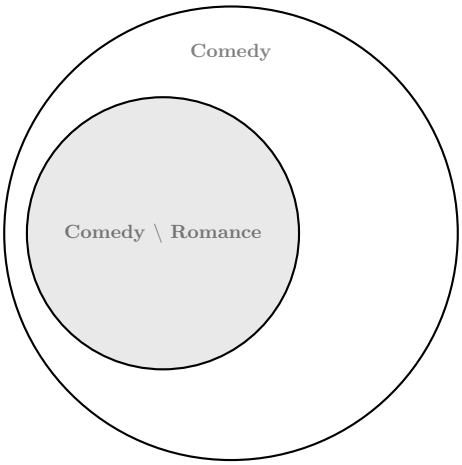

Figure 3: Sub-class definition

3.4.5. Additional Considerations on the Boolean Classes

Given the elementary boolean operations we presented, we can also create a new boolean class using the results of the previous operations, by combining them with further operations of the same type, e.g., $(nce_1 \vee nce_2) \wedge (nce_2 \neg nce_3)$.

It should be also noted that only the NOT operation, together with one of the other two operations (AND and OR) is enough to express all possible

combination of classes, as shown in Equation (7).

$$\begin{aligned} x \wedge y &= \neg(\neg x \vee \neg y) \\ x \vee y &= \neg(\neg x \wedge \neg y) \end{aligned} \tag{7}$$

3.5. User Segmentation

This step compares the output of the two previous steps (i.e., the set NCE of vectors related to the *neural class embeddings*, and the set M of vectors related to the *user models*), in order to infer which classes are relevant for a user. The main idea is to consider which layers are relevant for a user u (this information is stored in the user model m_u) and evaluate which classes are characterized by the layers in m_u (this information is contained in each vector nce_k, which contains which layers are relevant for the class c_k). The key concept behind this step is that *we do not consider the items a user evaluated anymore*. Each vector in NCE is used to estimate the relevance of each class for that user. Therefore, *a user might be associated to classes of items she never expressed a preference for, but characterized by layers that also characterize the user model*.

Since both the user models and the neural class embeddings are vectors, the most straightforward form of comparison (which is also the most employed in the literature) is to calculate the angle between them, through the cosine similarity. A user is associated to the segment of the class c_k, represented by the neural class embedding nce_k, if the cosine between the vectors (i.e., their similarity) is higher than a threshold φ, as shown in Equation (8).

$$cos(nce_k, m_u) \geq \varphi \tag{8}$$

This choice allows us to associate a user to more than a class, as it naturally happens in real-world scenarios, where a user can like more than one movie/music genre. Indeed, the vector representation of her preferences guides the association to the segments.

4. Experiments

This section describes the experiments performed to validate our proposal. In Section 4.1 we present the experimental setup and strategy, in Section 4.2 the dataset employed for the evaluation, Section 4.3 illustrates the involved metric, and Section 4.4 contains the results.

4.1. Experimental Setup and Strategy

The experiments have been performed using the Java language with the support of Java API implementation for word2vec, called *DL4J* (*Deep Learning for Java*)[13], and the real-world dataset Yahoo! Webscope Movie dataset (R4)[14].

[13] http://deeplearning4j.org/word2vec
[14] http://webscope.sandbox.yahoo.com

The experimental framework was developed by using a machine with an Intel i7-4510U, quad core (2 GHz $\times$ 4) and a Linux 64-bit Operating System (Debian Jessie) with 4 GBytes of RAM. To validate our proposal, we performed four sets of experiments:

1. **Neural Class Embeddings validation.** Our segmentation is based on a data structure that is built by combining the word embeddings of the items classified with it. To validate the effectiveness of this data structure, we divided the entire dataset into two parts (i.e., 90% was employed as the training set and 10% as the test set). We used the 10724 items in the training set to build the $NCEs$, and the 1191 items in the test set to verify the capability of the NCE models to correctly characterize the classes. In order to make this evaluation, we calculated the cosine similarity between the vector representation of the items in the test set and the NCEs of the classes.
2. **Analysis of the φ parameter.** The segmentation is built by putting together all the users with a similarity higher than a threshold φ with a class. This experiment analyzes the cardinality of the segments according to the different values of the parameter.
3. **Analysis of the primitive-class segments.** This experiment analyzes the segments of users associated to each primitive class, in order to evaluate the capability of our proposal to include also users who did not express explicit preferences for a class but might be interested in it.
4. **Analysis of the combined-classes segments.** This experiment analyzes the segments of users associated to each boolean class.

It should be observed that in order to validate the capability of our proposal to detect users who are not characterized by explicit preferences for a class, we compare with the so-called topic-based approach employed by both Google's AdWords and Facebook's Core Audiences. In order to do so, in the experiments number 3 and 4, we characterize the relevance of each class for a user, by considering how many movies of a genre a user evaluated (i.e., we are considering a scenario in which the topic of interest is a genre of movies, which is equivalent to our classes). This is done since the companies did not reveal how they associate users to topics, and in order to make a direct comparison between an approach that uses explicit preferences and our class-embedding-based approach.

4.2. Dataset

The used dataset contains a large amount of data related to users preferences expressed on the Yahoo! Movies community that are rated on the base of two different scales, from 1 to 13 and from 1 to 5 (we have chosen to use the latter). The training data is composed by 7,642 users ($\mid U \mid$), 11,915 movies/items ($\mid I \mid$), and 211,231 ratings ($\mid R \mid$). The average user rating ($\overline{R_u} = \frac{\sum_u \overline{r_u}}{|U|}$, *macro-averaged*) is 3.70 and the average item rating (*macro-averaged*) is 3.58. The average number of ratings per user is 27.64 and the average number of ratings per item is 17.73. All users have rated at least 10 items and all items

are rated by at least one user. The density ratio ($\delta = \frac{|R|}{|U|*|I|}$) is 0.0023, meaning that only 0.23% of entries in the user-item matrix are filled. As shown in Table 1, the items are classified by Yahoo in 19 different classes (movie genres), and it is should be noted that each item may be classified in multiple classes.

01	Action/Adventure	11	Musical/Performing Arts
02	Adult Audience	12	Other
03	Animation	13	Reality
04	Art/Foreign	14	Romance
05	Comedy	15	Science Fiction/Fantasy
06	Crime/Gangster	16	Special Interest
07	Documentary	17	Suspense/Horror
08	Drama	18	Thriller
09	Kids/Family	19	Western
10	Miscellaneous		

Table 1: Yahoo! Webscope R4 Genres

4.3. Metric

In order to detect the relevance of each class for the users in the topic-based segmentation we compare our approach to, we use the well-known *elbow method*. In other words, we increase the value of the number of evaluated-movies occurrences and calculate the variance, as shown in Equation (9), where x denotes the number of users involved, and n is the number of measures performed: at the beginning we can note a low level of variance, but at some point the level suddenly increases; following the *elbow method* we chose as threshold value the number of evaluated-movies occurrences used at this point.

$$S^2 = \frac{\sum(x_i - \overline{x})}{n-1} \tag{9}$$

4.4. Experimental Results

This section presents the results of each experiment previously presented.

4.4.1. Neural Class Embeddings validation

In this set of experiments we tested the capability of a NCE to model the characteristics of a class.

The results show that the NCE models are able to correctly classify the 78.50% of the items (i.e., the highest value of cosine similarity between an item and a class was with the class that classified the item).

Once we have verified the effectiveness of the NCE models, we have defined them by using the entire dataset.

4.4.2. Analysis of the φ parameter

A user is assigned to the segment of a class if the similarity with the class is higher than a threshold. Here, we will evaluate the cardinality of the generated segments according to the different values of the threshold φ. This will allow us to understand which values would lead to a significant analysis of the segments. Figure 4 reports in the x axis the different values of φ and in the y axis a value

called *Selectivity* that indicates when all the users are added to all the segments (100%) and when no user is added to no segment (0%).

The optimal range of threshold values to test is the interval where a variation in the similarity value leads toward a well-separated segmentation (i.e., without an excessive overlapping between classes) of the users among all the classes. In our case, the considered values are $\varphi = \{0.90, 0.92, \ldots, 0.98\}$, highlighted in the figure with a grey area.

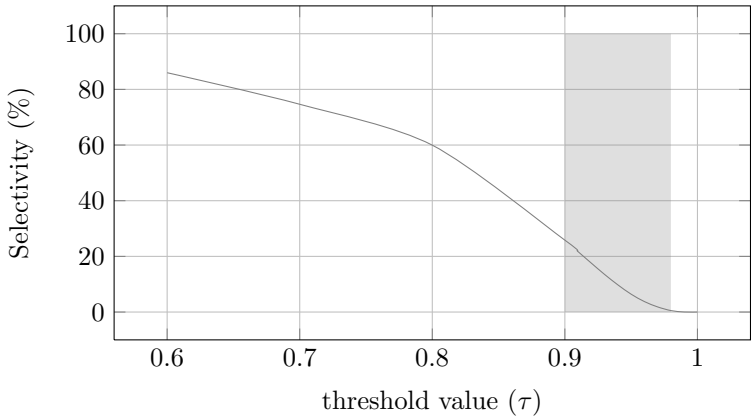

Figure 4: *Optimal threshold range*

4.4.3. Analysis of the primitive-class segments

Here we analyze the segments generated considering the NCEs produced for the primitive classes. In order to make a comparison with topic-based baseline approach, Table 2 first reports the number of preferences considered for each class in order to decide if a user should or should not be associated to the segment of that class.

In Table 3, we analyze the segments generated by our approach. As the results show, with all the threshold values our segments are bigger in size than those obtained with the baseline approach ("Native segment" column). This means that if we consider the explicit preferences given by the users and perform the so-called native segmentation when the number of preferences allows the approach to differentiate the segments, the result are segments composed by small amounts of users.

Class	*Elbow*	*Class*	*Elbow*
1	29	11	4
2	7	12	12
3	4	13	1
4	9	14	8
5	45	15	17
6	8	16	3
7	2	17	15
8	40	18	16
9	12	19	6
10	1		

Table 2: Native elbow values

Column "Relevant Users" indicates how many users that have been placed in a segment by our approach are actually related to the class (i.e., how many users positively evaluated items of that class). It should be observed that this extremely accurate result is obtained *without knowing in the segmentation process which items the user evaluated.*

Note that a result like the one presented in Table 3 would allow an advertiser to choose the threshold value by deciding between bigger or smaller segments of users to target.

Items	NCE	Relevant	Native	NCE	Relevant	Native	NCE	Relevant	Native	NCE	Relevant	Native	NCE	Relevant	Native
Class	$\varphi = 0.90$	Users	Segment	$\varphi = 0.92$	Users	Segment	$\varphi = 0.94$	Users	Segment	$\varphi = 0.96$	Users	Segment	$\varphi = 0.98$	Users	Segment
1	6003	5924	227	5266	5224	227	4174	4157	227	2545	2543	227	516	516	227
2	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
3	1760	1271	305	1062	837	305	458	397	305	87	85	305	3	3	305
4	862	476	64	331	198	64	77	51	64	10	8	64	0	0	64
5	3582	3563	124	2544	2537	124	1481	1479	124	551	550	124	51	51	124
6	4114	3712	345	3161	2877	345	2033	1881	345	847	807	345	67	67	345
7	3357	257	114	2163	162	114	926	81	114	92	11	114	0	0	114
8	2713	2663	126	1778	1762	126	961	957	126	357	357	126	41	41	126
9	176	172	124	41	41	124	7	7	124	1	1	124	0	0	124
10	0	0	11	0	0	11	0	0	11	0	0	11	0	0	11
11	19	17	156	13	13	156	8	8	156	4	4	156	1	1	156
12	2731	1564	110	1837	1071	110	800	545	110	84	69	110	0	0	110
13	0	0	25	0	0	25	0	0	25	0	0	25	0	0	25
14	1779	1607	306	1149	1066	306	589	560	306	203	198	306	25	25	306
15	1534	1513	205	831	825	205	347	345	205	73	73	205	4	4	205
16	229	14	21	113	10	21	36	4	21	7	3	21	0	0	21
17	3527	2238	90	1936	1389	90	591	502	90	87	83	90	13	13	90
18	3933	3641	227	2763	2601	227	1534	1469	227	487	472	227	37	36	227
19	1180	315	28	497	172	28	142	65	28	25	15	28	1	1	28

Table 3: Analysis of the segments

In order to summarize the results obtained with different threshold values, Figure 5 shows the average size of the partitions obtained by adopting our approach, and comparing it with the native partitioning, while Figure 6 reports the mean number of relevant users for each class.

As we can observe, the relevance of the additional users is proved for all classes except the *7th* (i.e., documentary), due to the difficulty to semantically characterize this heterogeneous class of items.

4.4.4. Analysis of the combined-classes segments

The last set of experiments is aimed at verifying the user segmentation generated through the combination of classes. In order to perform this operation we take into account the two classes with which most items were evaluated (i.e., 5 and 1). The classes were combined to generate an *Interclass-based NCE* and

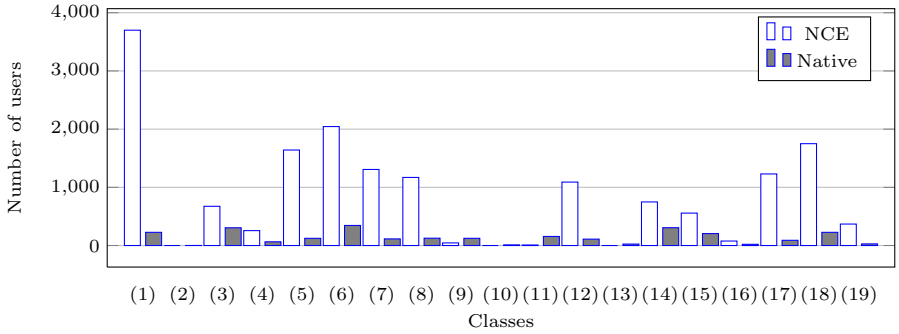

Figure 5: *Mean size of partitions*

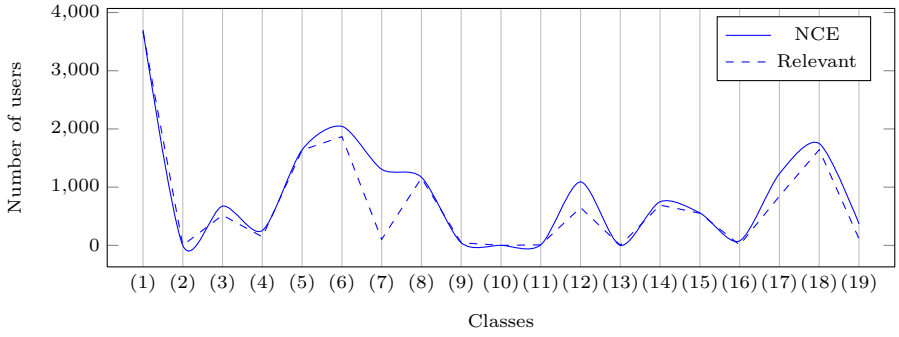

Figure 6: *Mean relevance of the NCE users*

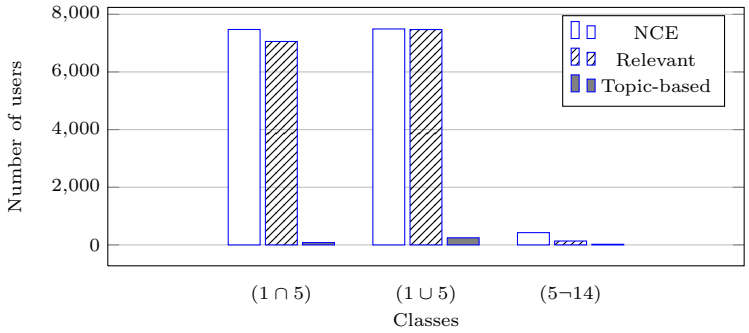

Figure 7: *Combined classes*

a *Superclass-based NCE*. We also test a *Subclass-based NCE* by taking into account the classes 5 and 14, i.e., the ones in the dataset that have been most used to co-classify the items.

Figure 7 shows the mean number of users placed in the segments by using a threshold value $\varphi = \{0.1, 0.2, \ldots, 0.6\}$ (above these values no segmentation is possible). These results show that also when the classes are combined, they generate large segments of users to target and these users have shown interest in the classes. This means that a NCE is an effective data structure to model the classes, even when they are combined.

5. Conclusions and Future Work

This paper presented an approach to segment the users by analyzing the items positively evaluated by them, in order to consider reliable user preferences. These items were analyzed by extracting the word embeddings and by building a novel type of class model, named *Neural Class Embedding*. The models allowed us to understand what characterizes each class of items and to generate segments of users with certain characteristics. In this way, we designed an approach that does not generate trivial segments, since the segmentation process is not purely based on the items evaluated by the users. Moreover, the segmentation is easily interpretable, as the advertisers are only required to specify the class (or classes) of items they want to target, which can be combined through an effective boolean algebra presented in this study. Future work will test the capability of our approach to characterize segments of users whose purchased items are semantically related. This approach would allow us to target the users in a different way, e.g., by performing group recommendations to them (i.e., by recommending items to groups of "semantically similar" users).

Acknowledgments

This work is partially funded by Regione Sardegna under project NOMAD (Next generation Open Mobile Apps Development), through PIA - Pacchetti

Integrati di Agevolazione “Industria Artigianato e Servizi” (annualità 2013), and by MIUR PRIN 2010-11 under project “Security Horizons”.

References

[1] X. Gong, X. Guo, R. Zhang, X. He, A. Zhou, Search behavior based latent semantic user segmentation for advertising targeting, in: Data Mining (ICDM), 2013 IEEE 13th International Conference on, 2013, pp. 211–220.

[2] S. Tu, C. Lu, Topic-based user segmentation for online advertising with latent dirichlet allocation, in: Proceedings of the 6th International Conference on Advanced Data Mining and Applications - Volume Part II, ADMA'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 259–269.

[3] A. Spink, B. J. Jansen, D. Wolfram, T. Saracevic, From e-sex to e-commerce: Web search changes, Computer 35 (3) (2002) 107–109.

[4] S. Y. Rieh, H. I. Xie, Analysis of multiple query reformulations on the web: The interactive information retrieval context, Inf. Process. Manage. 42 (3) (2006) 751–768.

[5] P. Boldi, F. Bonchi, C. Castillo, S. Vigna, From "dango" to "japanese cakes": Query reformulation models and patterns, in: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 183–190.

[6] G. Armano, A. Giuliani, E. Vargiu, Semantic enrichment of contextual advertising by using concepts, in: J. Filipe, A. L. N. Fred (Eds.), KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Paris, France, 26-29 October, 2011, SciTePress, 2011, pp. 232–237.

[7] G. Armano, A. Giuliani, E. Vargiu, Studying the impact of text summarization on contextual advertising, in: F. Morvan, A. M. Tjoa, R. Wagner (Eds.), 2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011, IEEE Computer Society, 2011, pp. 172–176.

[8] R. Saia, L. Boratto, S. Carta, Semantic coherence-based user profile modeling in the recommender systems context, in: Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2014, Rome, Italy, October 21-24, 2014, SciTePress, 2014, pp. 154–161.

[9] R. D. Burke, M. Ramezani, Matching recommendation technologies and domains, in: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 367–386.

[10] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, in: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 73–105.

[11] C. Gustav Johannsen, Understanding users: from man-made typologies to computer-generated clusters, New Library World 115 (9/10) (2014) 412–425.

[12] S. C. Bourassa, F. Hamelink, M. Hoesli, B. D. MacGregor, Defining housing submarkets, Journal of Housing Economics 8 (2) (1999) 160 – 183.

[13] A. Nairn, P. Bottomley, Something approaching science? cluster analysis procedures in the crm era, in: H. E. Spotts (Ed.), Proceedings of the 2002 Academy of Marketing Science (AMS) Annual Conference, Developments in Marketing Science: Proceedings of the Academy of Marketing Science, Springer International Publishing, 2003, pp. 120–120.

[14] S. Dolnicar, K. Lazarevski, Methodological reasons for the theory/practice divide in market segmentation, Journal of Marketing Management 25 (3-4) (2009) 357–373.

[15] S. Dibb, L. Simkin, A program for implementing market segmentation, Journal of Business & Industrial Marketing 12 (1) (1997) 51–65.

[16] T. Mikolov, Q. V. Le, I. Sutskever, Exploiting similarities among languages for machine translation, CoRR abs/1309.4168.

[17] N. Djuric, H. Wu, V. Radosavljevic, M. Grbovic, N. Bhamidipati, Hierarchical neural language models for joint representation of streaming documents and their content, in: Proceedings of the 24th International Conference on World Wide Web, WWW '15, ACM, New York, NY, USA, 2015, pp. 248–255.

[18] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, Vol. 32 of JMLR Proceedings, JMLR.org, 2014, pp. 1188–1196.

[19] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., 2013, pp. 2787–2795.

[20] R. Socher, D. Chen, C. D. Manning, A. Y. Ng, Reasoning with neural tensor networks for knowledge base completion, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information

Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., 2013, pp. 926–934.

[21] R. Kiros, R. S. Zemel, R. R. Salakhutdinov, A multiplicative model for learning distributed text-based attribute representations, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 2348–2356.

[22] R. Kiros, R. Salakhutdinov, R. S. Zemel, Multimodal neural language models, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, Vol. 32 of JMLR Proceedings, JMLR.org, 2014, pp. 595–603.

[23] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, ACM, New York, NY, USA, 2014, pp. 701–710.

[24] G. Armano, E. Vargiu, A unifying view of contextual advertising and recommender systems, in: A. L. N. Fred, J. Filipe (Eds.), KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Valencia, Spain, October 25-28, 2010, SciTePress, 2010, pp. 463–466.

[25] A. Addis, G. Armano, A. Giuliani, E. Vargiu, A recommender system based on a generic contextual advertising approach, in: Proceedings of the 15th IEEE Symposium on Computers and Communications, ISCC 2010, Riccione, Italy, June 22-25, 2010, IEEE, 2010, pp. 859–861.

[26] E. Vargiu, A. Giuliani, G. Armano, Improving contextual advertising by adopting collaborative filtering, ACM Trans. Web 7 (3) (2013) 13:1–13:22.

[27] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, Z. Chen, How much can behavioral targeting help online advertising?, in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, ACM, New York, NY, USA, 2009, pp. 261–270.

[28] H. Beales, The value of behavioral targeting, Network Advertising Initiative.

[29] Y. Chen, D. Pavlov, J. F. Canny, Large-scale behavioral targeting, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 209–218.

[30] J. Bian, A. Dong, X. He, S. Reddy, Y. Chang, User action interpretation for online content optimization, IEEE Trans. on Knowl. and Data Eng. 25 (9) (2013) 2161–2174.

[31] Z. Yao, T. Eklund, B. Back, Using som-ward clustering and predictive analytics for conducting customer segmentation, in: Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, ICDMW '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 639–646.

[32] Y. K. Zhou, B. Mobasher, Web user segmentation based on a mixture of factor analyzers, in: Proceedings of the 7th International Conference on E-Commerce and Web Technologies, EC-Web'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 11–20.

[33] X. Wu, J. Yan, N. Liu, S. Yan, Y. Chen, Z. Chen, Probabilistic latent semantic user segmentation for behavioral targeted advertising, in: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising, ADKDD '09, ACM, New York, NY, USA, 2009, pp. 10–17.

[34] J. Mazanee, Market Segmentation, in: J. Jafari (Ed.), Encyclopedia of Tourism, London: Routledge, 2000.

[35] S. Dolničar, Beyond "commonsense segmentation": A systematics of segmentation approaches in tourism, Journal of Travel Research 42 (3) (2004) 244–250.

[36] J. H. Myers, E. M. Tauber, Market Structure Analysis, American Marketing Association, 1977.

[37] M. Wedel, W. A. Kamakura, Market Segmentation: Conceptual and Methodological Foundations (International Series in Quantitative Marketing), Kluwer Academic Publishers, 2000.

[38] S. A. Munson, P. Resnick, Presenting diverse political opinions: How and how much, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, ACM, New York, NY, USA, 2010, pp. 1457–1466.

[39] E. Pariser, The Filter Bubble: What the Internet Is Hiding from You, Penguin Group , The, 2011.

[40] L. Festinger, A theory of cognitive dissonance, Vol. 2, Stanford university press, 1962.

[41] S. Park, S. Kang, S. Chung, J. Song, Newscube: delivering multiple aspects of news to mitigate media bias, in: Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009, ACM, 2009.

[42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., 2013, pp. 3111–3119.