Exploiting Social Judgements in Big Data Analytics

Christoph Lofi, Philipp Wille

Institut für Informationssysteme
Technische Universität Braunschweig
38106 Braunschweig, Germany
{lofi, wille}@ifis.cs.tu-bs.de

Abstract. Social judgements like comments, reviews, discussions, or ratings have become a ubiquitous component of most Web applications, especially in the e-commerce domain. Now, a central challenge is using these judgements to improve the user experience by offering new query paradigms or better data analytics. Recommender systems have already demonstrated how ratings can be effectively used towards that end, allowing users to semantically explore even large item databases. In this paper, we will discuss how to use unstructured reviews to build a structured semantic representation of database items, enabling the implementation of semantic queries and further machine-learning analytics. Thus, we address one of the central challenge of Big Data: making sense of huge collections of unstructured user feedback.

Keywords: Big Data; User Generated Content; Data Mining; Latent Semantics

1 Introduction

The recent years have brought several changes in how the Web is used by both individual users and companies alike. Especially, the Social Web had a strong impact and has now become a major innovator of technology. Users got accustomed to an active and contributive usage of the Web, and feel the need to express themselves and connect with like-minded peers. As a result, social networking sites like Facebook amassed over 940 million active users. At the same time, there are countless special-interest sites for music, movies, art, or anything that is of interest to any larger group. But the real revolution lies in the way people interact with these sites: Following their social nature, millions of people discuss, rate, tag, review, or vote content and items they encounter on the Web. Therefore, "I Like" buttons, star scales, or comment boxes are omnipresent

 In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published a http://ceur-ws.org

on today's Web. Of course, recognizing the value of the exploitation of such activities, many companies encourage the creation of such user-generated feedback [1] and exploit it in order to analyze their user base, provide better meta-data and user interaction, or to optimize their marketing strategies. Storing and querying the huge amount of data related to these socially-driven Web activities and also supporting the subsequent analysis are among the central concerns in the current discussions about Big Data and Cloud Computing systems. From a database research point of view, there is a clear challenge given by Big Data applications: huge amounts of data need to be stored and served efficiently and flexibly in a distributed fashion. This results in many interesting database-like systems which have to decide on tough trade-offs with respect to possible database features, efficiency, and scalability, e.g., [2]. However, beyond storage, there is another at least equally challenging problem: How can all that data, and especially user-generated judgements and feedback, be put to a practical use? Here, a core problem is that user contributions in the Social Web are often very hard to control and usually do not follow strict schemas or guidelines. For example, a user finding an interesting online news article might vote for that article on her preferred social site, while a user leaving the cinema after a particular bad movie experience may log onto her favorite movie database, rating the movie lowly, and venting her disappointment in a short comment or a more elaborate review.

In this paper, we discuss the challenge of building structured, but latent representations of "experience items" stored in a database (like movies, books, music, games, but also restaurants or hotels) from unstructured user feedback. Such representations should encode the consensual perception of an item from the perspective of a large general user base. If this challenge could be solved, established database techniques like SQL-queries, similarity queries, but also several data mining techniques like clustering could be easily applied to user-generated feedback. In the following, we will use movies as an example use case. However, the described techniques can easily be transferred to any other domain which has user ratings or reviews available. In detail, our outline is:

- We explain our perceptual space, an established state-of-the-art latent representation of items based on user-item ratings. While this technique is accepted and proven, the required data is hard to obtain and carries some privacy concerns.
- Instead of using ratings, we discuss how to use user-provided natural language reviews to derive a semantically meaningful item representation. As reviews are easier to obtain, they could serve as an attractive alternative to rating-based approaches. We evaluate three different approaches and compare them to an established rating-based approach. Especially, we will focus on the use of neural language embeddings, a currently emerging technique from the natural language processing community.
- As our evaluation will show, there are still quality problems with directly turning review texts into latent item representations. We will discuss the potential sources of these problems, and will outline possible solutions and remedies to be explored by later research. Furthermore, we will briefly discuss the challenge of making some of the latent dimensions explicit and explainable.

2 Experience Items, User Content and Latent Representations

In this paper, we use an e-commerce scenario where users can browse and buy frequently consumable experience. This scenario is a prime application for user-generated judgements: user-friendly interaction with experience items is notoriously difficult, as there is an overwhelming number of those items easily available, some of them being mainstream, vastly popular, and well-known, but most of them being relatively unknown long tail products which are hard to discover without suitable support. Even more, the subjective user experience those products will entail (which, for most people, is the deciding factor for buying the product) are difficult to describe by typically available meta-data like production year, actor names, or even rough genre labels. Due to this problem, web services dealing with experience products enthusiastically embraced techniques for motivating the creation of user-generated judgements in the form of ratings, comments or reviews. In its most naïve (but very common) implementation, rating and review data are simply displayed to users without any additional processing. Querying and discovering items still relies on traditional SQL-style queries and categorizing based on non-perceptual meta-data (e.g., year, actor list, genre label, etc.). Manually ingesting these user judgements may help potential new customers to decide if they will like or dislike a certain item, but it does not really help them to discover new items beyond their expertise (i.e., this approach works fine if a user knows exactly what she is looking for, but has not yet come to a final buying decision). This led to the development of recommender systems [3, 4], which proactively predict which items a user would enjoy. Often, this relies on collaborative filtering techniques [3] which exploit a large number of user-item ratings for predicting a user's likely ratings for each yet-unrated item. While collaborative filtering recommender systems have been proven to be effective [5], they have only very limited query capabilities (basically, a recommender system is just a single static query for each user).

For enabling semantic queries like similarity exploration [6], the first step is to find semantically meaningful representations of database items going beyond available structured meta-data. It has been shown that experience items are generally better characterized by their *perceived properties*, e.g. their mood, their style, or if there are certain plot elements – information which is rarely explicitly available and expensive to obtain.

Therefore, we aim at extracting the most relevant perceptual aspects or attributes describing each item from user-generated judgements automatically, resulting in a vector representing these attributes. Some existing systems like Pandora's Music Genome [7] and Jinni's Movie Genome [8] already worked on this challenge, but these systems are proprietary and rely on strong human curation and expert taxonomies. In contrast, we try to use fully automatic approaches. Therefore, we investigate *dense latent representations* of each item, i.e. for each item, we mine all values with respect to each perceptual attribute. However, while these attributes might have a real-world interpretation, that interpretation is unknown to us (for example, one attribute might represent how scary a movie is, but this attribute will simply have a generic name and we do not know that it indeed refers to scariness). Basically, when creating a dense latent representation, each item is embedded in a high-dimensional vector space (therefore, such

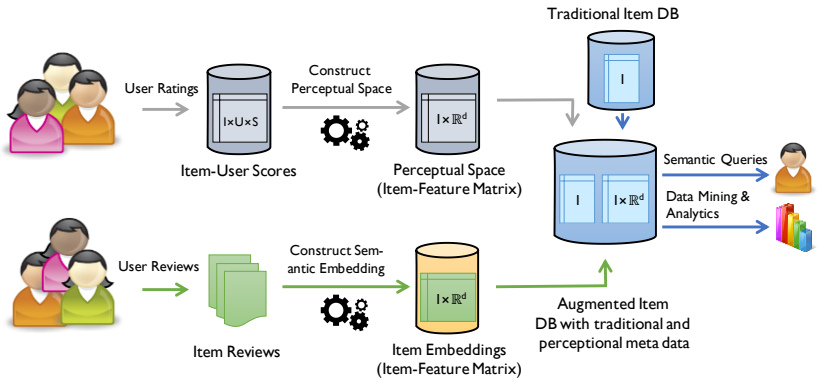

Figure 1: Augmenting Item Meta Data with User Generated Information
Extracted latent vector representations can be used side-by-side for supporting, e.g., similarity queries or different data mining techniques like clustering or automatic labeling

techniques are also sometimes called "embeddings") with usually 100-600 automatically created dimensions where each dimension represents an (unlabeled) perceptual aspect (like scariness, funniness, quality of special effects, or even the presence of certain plot elements like "movie has slimy monsters").

Even without explicitly labeling the dimensions, latent representations can already provide tremendous benefits with respect to the user experience. They can directly be used by most state-of-the art data analytics and machine learning algorithms like clustering, supervised labeling, or regression. Also, from a user's perspective, such representations can be used with great effect to allow for semantic queries as we have shown in [6] for movies. Here, having a meaningful implementation for measuring the semantic similarity (derived from the vector distance between movies) has been exploited to realize personalized and user-friendly queries using the query-by-example (QBE) paradigm. In that work ([6]), we relied on Perceptual Spaces, a latent representation derived from a large collection of user-item ratings which we will use as a reference implementation in this paper. Unfortunately, such ratings are hard to obtain and also come with some privacy concerns. Therefore, a core contribution of this paper is exploring alternative techniques for building item embeddings using much more accessible reviews (see section 2.2) instead. The resulting overall workflow of our aproach is summarized in figure 1.

2.1 Perceptual Spaces: Latent Representations from Ratings

There are several (somewhat similar) techniques for building latent semantic representations based on rating data, which mostly differ with respect to the chosen basic assumptions and decomposition algorithms. Our Perceptual Spaces introduced in [9] and [6] rely on a factor model using the following assumptions:

Perceptual Spaces use the established assumption that item ratings in the Social Web are a result of a user's preferences with respect to an item's attributes [10]. Using movies as an example, a given user might have a bias towards furious action; therefore, she

will see movies featuring good action in a slightly more positive light than the average user who cares less for action. The sum of all these likes and dislikes, combined with a user's general rating bias and the consensual quality of a movie will lead to the user's overall perception of that movie, and will therefore ultimately determine how she rates it on a social movie site. The challenge of perceptual spaces is to reverse a user's rating process: For each item which was rated, commented, or discussed by a large number of users, we approximate the actual characteristics (i.e., the systematic bias) which led to each user's opinion as numeric features. This process usually only works if there is a huge number of ratings available, with each user rating many items and each item being rated by many users (this does of course also apply to all other techniques using user-item ratings for latent representations or recommendations). The Perceptual Space is then a consensual view of the item's properties from the perspective of the average user, and one can claim that it therefore captures the "essence" of all user feedback. A similar reasoning is also successfully used by other latent, e.g. [5, 11].

As our experiments in [9] showed, quality of perceptual spaces increase with the involvement and activity of users: rating data obtained from a restaurant data set (where users in a large "lazy" community rated only few restaurants each) produced worse results than using more active Netflix users. Very strong results could be achieved using an enthusiast community focused on discussing board games, here an even smaller group of highly active users rate a huge collection of board games each.

In the experiments presented in section 3, we rely on the dataset released during the Netflix Prize challenge [4] in 2005 (as an alternative, the MovieLens dataset [12] could be used which contains fewer ratings for a larger number of more recent movies). The Netflix dataset is still one of the largest user-item-rating datasets available to the research community. This fact is also the central problem limiting the value of rating-based approaches. While large web companies like Amazon, Google, or Netflix have large user-item rating datasets available in-house, these datasets are usually neither accessible nor shared. One reason for this problem is that it is very hard to foster a community active and large enough to reliably provide a huge number of ratings, and therefore companies which were able to overcome these challenges often consider their rating data as valuable business assets which are kept protected. Furthermore, user ratings can be problematic from a privacy perspective: as shown by [13], this type of rating data can be de-anonymized surprisingly well, opening legal concerns for sharing rating datasets. In fact, there have indeed been problems with bad publicity and legal issues with respect to de-anonymizing the Netflix dataset after its release, e.g., [14]. But even in a closed in-house environment, the possibility of de-anonymization and user profiling fosters discomfort in the user base – and users are less motivated to actively contribute if they feel that their privacy could be compromised [15]. In contrast, reviews are clearly public, so users are not surprised (and angry) by the fact that somebody reverse-engineered their seemingly anonymous rating. Furthermore, if datasets are shared, all references to actual users can be fully removed (in rating data sets, user ids can only be obfuscated, but not removed entirely. Reversing this obfuscation is the core of de-anonymization techniques.)

2.2 Neural Word Embeddings: Latent Representations from Reviews

In the last section, we argued that obtaining the rating data required for building latent representations of items can be very challenging. Therefore, in this section we introduce latent representations based on reviews. A good review dataset is significantly easier to create and share: it is acceptable if users have only a brief period of activity (e.g., writing only few reviews and then turn inactive again) as long as there are enough reviews overall (in contrast, rating-based approaches usually can only consider ratings from users who have rated many different items). Furthermore, reviews can be anonymized effectively.

Using reviews to build semantic representations seems to be an alluring idea: in a good review, a user will take the time to briefly summarize the content of an item, and then expresses her feelings and opinions towards it. Transforming the essence of a large number of reviews into a latent representation of items promises to be a semantically valuable alternative to ratings. In this paper, we will discuss latent fixed-length vector representations for this task. These approaches have the advantage that they are particularly easy to use in machine learning algorithms, and can naïvely be utilized to measure similarity between items which benefits explorative queries. As an alternative route, one could also use opinion mining techniques which use additional natural language analysis techniques to explicitly extract features and opinions from texts (see [16]). Here, the core challenge lies in how to match the extracted features into a uniform representation, which is a problem we will investigate in a later work.

In the following, we discuss three different fixed-length approaches for creating latent item representations from reviews. They share a similar core workflow: a) first, we represent a single review as a fixed-length feature vector, and then we b) combine all review vectors of a given item into a single latent item representation. In this work, we use the centroid vector of all review vectors for combining.

The simplest but due to its surprising efficiency and accuracy still very popular technique for representing a given text (e.g., a review) as a fixed length vector is the bag-of-words model (BOW) [17]. In its basic version, the BOW model counts the number of occurrences of each term/word in a document, and each document in a given collection is represented by a word count vector with all words in the whole collection (the vocabulary) as dimensions. It is an accepted assumption that most machine learning tasks work better when term weightings are used instead of simple word counts. We therefore use the popular TF-IDF weighting scheme [18], in which words commonly appearing in documents have generally a lower weight than specific words appearing only in few documents. As a preprocessing step, we also remove all stop words (i.e., words without any particular semantics for the purpose of latent representation).

As a result, most of these TF-IDF document vectors will be very sparse as each document only covers a fraction of vocabulary words. In many real life tasks, dense vector representations have been shown to achieve better results (as, e.g., in [19] for semantic clustering). The core idea of many dense vector representations is to apply dimension reduction techniques to the matrix of all document vectors M, reducing it to its most dominant (latent) dimensions (usually around 100 to 600 dimensions). This process usually relies on matrix decomposition, i.e. the matrix M is decomposed into at two

matrixes with a significantly reduced number of latent dimensions such that their product approximates M as closely as possible. This can be achieved by approaches like principal component analysis (PCA) [20], or latent semantic analysis (LSA) [21]. In our evaluations, we will apply LSA to the TF-IDF representation.

In the last few years there has been a surge of approaches proposing to build dense word vectors not by using matrix factorization, but by using neural language models which have the training of a neural network at their core. Early neural language models were designed to predict the next word given a sequence of initial words of a sentence [22] (as for example used in text input auto-completion) or to predict a nearby word given a cue word [23]. While neural language models can be designed for different tasks and trained with a variety of techniques, most share the trait that they internally create a dense vector representation of *words* (note: not documents!). This representation is often referred to "neural word embeddings". The usefulness of these embedding may vary with respect to the chosen tasks, but it has been shown that they have surprising (and hard to explain) properties when it comes to modelling the semantics and perceived similarities of words (like being able to represent rhetoric analogies [25]). The common process of training a neural language model is to learn real-valued embeddings for words in a predefined vocabulary. In each training step, a score for the current training example is computed based on the embeddings in their current state. This score is compared to the model's objective function, and then the error is propagated back to update both the model and the weights. At the end of this process, the embeddings should encode information that enables the model to optimally satisfy its objective [26].

Early approaches like [22] used rather slow multi-layer neural networks, but current approaches adopted a significantly simpler technique using non-linear hidden layer neural networks (like the popular skip-gram negative sampling approach (SGNS) [23, 27]). These models are trained using 'windows' extracted from a natural language corpus (i.e. an unordered set of words which occur nearby in a text sequence in the corpus). The model is trained to predict, given a single word from the vocabulary, those words which will likely occur nearby it (i.e. share the same windows).

Most neural language models focus on vector representations of single words. In order to represent a whole review as a latent vector, we will use a novel neural document embedding technique described in [28], a multi-word extension of the skip-n-gram model introduced in [23]. This technique brings several unique new features. In contrast to BOW or simply applying LSA, neural document embeddings have a sense of similarity between different words occurring in a text: e.g., in BOW-models words like "funny", "amusing", and "horrifying" are treated as equidistant in their semantics, while in reality "funny" and "amusing" are semantically very close. Another new feature of document embeddings is that they will consider the order of words in texts, i.e. all BOW-based approaches and also simple aggregates of neural word embeddings will create the same latent representation of a document regardless of the word order. In [28], it has been shown that these new features result in an tremendous increase of result quality for different semantic tasks like sentiment analysis, classification, or information retrieval. Therefore, we assume that using document embeddings will also result in an increase of quality when creating latent item representations in a Big Data environment.

3 Evaluation

In the following, we evaluate different review-based embeddings in comparison with our rating-based perceptual space [9] as a baseline. As the rating data for building a perceptual space can be hard to get, the following experiments investigate how well latent representation of movies mined from reviews can be used as replacements for the perceptual space. Our perceptual space is built from the Netflix dataset [4] which consists of 103M ratings provided by 480k users on 17k video and movie titles (all titles from 2005 and older). We filtered out all TV series and retained only full feature movies for our evaluation, leaving 11,976 movies. The initial construction of the 100-dimensional space took slightly below 2 hours on a standard notebook computer.

For the latent representations built from reviews, we used the Amazon Review dataset introduced in [29]. The full review dataset consists of 143.7 million reviews from May 1996 up to July 2014, covering all items sold at Amazon.com. We only used the "Movies and TV" category, leaving 64,835 products and 294,333 reviews. We applied each of the three review-based techniques described in section 2.2 to this corpus (the simple TF-IDF model, a standard LSA model using the previous TF-IDF with varying dimensionality, and a neural document embedding model [28] also with varying dimensionality). After the model generation, for the final experiments, we dropped all items with less than 20 reviews, and all reviews which have less than 2,000 characters (or 500 alternatively). (a similar cleaning procedure was also applied by Netflix to rating-based dataset, dropping items with very few ratings). Finally, we consider only movies which are in the Netflix dataset and which we also could reliably match to the Amazon review dataset by exact title matches. For many titles this is not easily possible as there is no uniform naming scheme across datasets (e.g., "Terminator 2: Judgement Day" vs. "Terminator 2: Ultimate Edition" - both referring to the same movie), and overall data is very dirty and ambiguous. To a certain extent, a better matching would have to rely on manually comparing the movie cover arts as this is the only meta-data available in both datasets besides the (ambiguous) title name. This finally leaves us with 3,284 movies, and each of these movies has an average of 8.58 reviews. With respect to computation time (using a standard notebook computer), building the BOW model took us 67 minutes, the LSA model 28 hours, and the neural embeddings take roughly 2 hours.

We consider this paper as a work-in-progress report of our current research efforts, and in the following, we will focus on the performance of the aforementioned approaches with respect to similarity computation, i.e. we will evaluate if the similarity measured between all pairs of movies in one of the review-based models correlates (using Spearman rank correlation) with the measured similarity of the same movies in the perceptual space. On one hand, we choose this evaluation design because all four different latent representation will likely choose different dimensions (including different dimensionalities), so vectors cannot be compared directly. On the other hand, we do not require the vector spaces to be equivalent as we only need them to behave comparably in application, and for most applications being able to measure item similarity is the only required feature for, e.g., explorative queries and cluster analysis. An in-depth evaluation of other aspects will follow at a later stage.

All in all, the measured correlations paint a discouraging picture at first: the correlation coefficient of the different techniques is rather low varying between 0.05 (BOW) and 0.30 (Neural Embedding). Considering the really low performance of BOW, the neural embeddings fared surprisingly well, as shown in figure 2. In figure 3, we show the correlation for neural embeddings with varying dimensions, and a minimal text length of 2,000 characters (as used in the experiments in figure 2), and a smaller minimal text length of 500 characters. Interestingly, the correlation increases when less dimensions are chosen. This is likely due to the fact that the neural model has a higher degree of abstraction with a lower number of dimensions. Also, as expected, result correlation decreases when a smaller minimum text length is chosen.

4 Issues and Outlook

As we have shown in the last section, similarity measured using latent representations built from reviews do not convincingly correlate with similarity in perceptual spaces. However, it is unclear what this low correlation means for practical applications: To our current knowledge, there is no study which examined how well rating-based representations like our perceptual space approximate real user perceptions from a psychological perspective in a quantitative way. We still used the perceptual space as a baseline because of the popularity of such item-rating based approaches, and their effectiveness has been shown in actual systems on many occasions (i.e. it is unclear in how far rating-based approaches are indeed "correct" and "complete"; however, they "work well", e.g. see [6]). Now, it could be possible that review-based representations are still semantically meaningful, but simply focus on different aspects of items: i.e. for ratings, people simply provide an overall judgement while in reviews, often certain dominant aspects are highlighted and discussed. This should indeed lead to different but still correct similarity semantics. In order to shed light on this problem, we would need to perform a study with a large group of users focusing on the performance and correctness of systems using either representation, which definitely will be a part of a later research

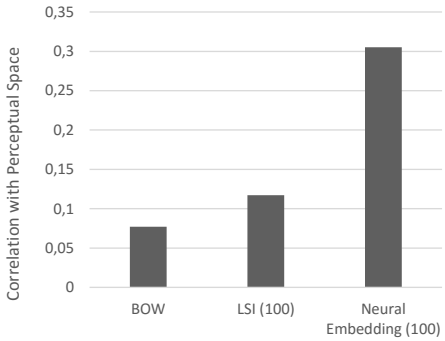

Figure 2: Correlation between review-based techniques and Perceptual Space
(number of dimensions in parenthesis)

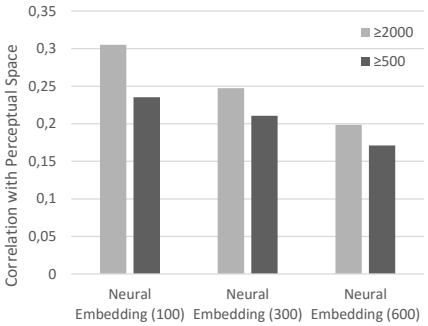

Figure 3: Neural Word Embedding with varying number of dimensions and minimal text length
(number of dimensions in parenthesis)

work. In any case, the workflow described in this paper leaves room for improvement (like introducing proper data cleaning), and we will discuss such issues below.

Data Cleaning and Review Quality: We manually inspected a selection of movies and their supposedly most similar titles as suggested by the neural document embedding technique. Here, it turned out that there are indeed some good matches in the similarity list, both confirmed by the perceptual space and the authors. However, there are also some random-looking titles suggested (e.g., for the movie "Terminator 2", both "Robocop" (a good match) and "Dream Girls Private Screenings" (a surprisingly bad match). The reason for this irritating behavior seems to be that there are many "bad" reviews (as for example most of the reviews of the second movie). "Bad" reviews are not discussing the movie itself, but other issues and do therefore not contribute to a meaningful representation. Typical examples are "I had to wait 5 weeks for delivery of the item! Stupid Amazon!", "Srsly?! Package damaged on delivery?", "I ordered the DVD version, got the Blue Ray!". For "Dream Girls", reviewers seem to be mostly concerned with the bad quality of the DVD version in comparison to the older VHS release. A similar thing happens in several reviews of the original DVD release of Terminator 2. Therefore, a next step would be excluding all reviews which do not discuss the content of the movie per se, but have other topics (e.g., print quality, delivery time, quality of customer service, etc.). However, this task is not trivial. It could be realized by training a machine classifier detecting topics, or by generic topic-modelling techniques like LDA [30]. This quality problem does not occur with our rating data, as in Netflix, it was made clear that users are supposed to rate only the content of a movie.

Overall, it seems that Amazon reviews are of rather mediocre detail and quality. In contrast, there are some online enthusiast communities like the aforementioned board game community which mostly consists of highly motivated members. There, user reviews are usually quite detailed and verbose, and it could be that our approach will yield significantly stronger results in that scenario. Another factor to consider is how we train our neural embedding models: we only used the Amazon review corpus for training. However, it is quite possible (and likely) that overall accuracy could be increased by also incorporating common knowledge corpora like the popular Wikipedia or Google News dumps [23] in the training process as this should result in better word sense semantics, which in turn should also benefit the representation of a whole review.

Additionally, we experimented only with one techniques for combining review vectors. Instead of simply computing an average vector, a weighted combination could improve quality considerably. In this sense, we also tried to combine reviews before computing the vector representations. However, this approach has prohibitive runtimes for training the document embedding, and we therefore stopped investigating it further.

Latent Representations and Explicit Properties: While latent representations of items can be used in a variety of machine learning tasks and can also be used for example-based user queries, we have no explicit real-world interpretation of the semantics of the different latent attributes. In [9], we have shown that certain perceived properties (like the degree of funniness) can be made explicit with only minimal human input using crowdsourcing-based machine regression. The core idea is that by providing few examples of items strongly exhibiting an interesting trait, and a few items which do not exhibit that trait at all, this trait can be approximated also for all other items as long as

the trait is somewhere covered in a combination of attributes of our latent representation. In our future work, we will focus on the challenge of how to find interesting traits automatically and how to minimize the required human input, which could either rely on opinion mining [16] or user-generated item tags [31].

5 Summary

In this paper, we discussed building dense sematic vector representations of database items from user-provided judgements. These representations can be used in many different applications like semantic queries and a multitude of data analytics tasks. Especially, we focused on neural language models, a new emerging technique from the natural language processing community which has shown impressive semantic performance for a wide variety of language processing problems. We compared how these review-based approaches compare to an established state-of-the-art rating-based technique using similarity measurements as a benchmark. Unfortunately, the results are less conclusive than we hoped for. Basically, measurements based on neural document embeddings correlate only weakly with those from our baseline. However, a brief qualitative inspection into the results reveal some obvious shortcomings of our current approach which will be fixed in future works. Especially, a central problem of review-based approaches in general seems to be low review quality, i.e. many reviews are off-topic or simply uninformative. Categorizing, filtering and weighting different reviews, among some other optimizations, should yield significantly better results in future works. In general, we can conclude that it is significantly more challenging to extract latent semantic attributes from reviews than from ratings, and therefore this challenge requires additional study.

References

1. Liu, Q. Ben, Karahanna, E., Watson, R.T.: Unveiling user-generated content: Designing websites to best present customer reviews. Bus. Horiz. 54, 231–240 (2011).
2. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst. 26, (2008).
3. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. 7, 76–80 (2003).
4. Bell, R.M., Koren, Y., Volinsky, C.: All together now: A perspective on the Netflix Price. CHANCE. 23, 24–24 (2010).
5. Koren, Y., Bell, R.: Advances in Collaborative Filtering. Recommender Systems Handbook. pp. 145–186 (2011).
6. Lofi, C., Nieke, C.: Exploiting Perceptual Similarity: Privacy-Preserving Cooperative Query Personalization. Int. Conf. on Web Information System Engineering (WISE). , Thessaloniki, Greece (2014).
7. John, J.: Pandora and the music genome project. Sci. Comput. 23, 40–41 (2006).
8. Jinni Movie Genome, http://www.jinni.com/discovery/.

9. Selke, J., Lofi, C., Balke, W.-T.: Pushing the Boundaries of Crowd-Enabled Databases with Query-Driven Schema Expansion. Proc. VLDB. 5, 538–549 (2012).
10. Kahneman, D., Tversky, A.: Psychology of Preferences. Sci. Am. 246, 160–173 (1982).
11. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. 22, 89–115 (2004).
12. Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: MovieLens unplugged: experiences with an occasionally connected recommender system. Int. Conf. on Intelligent User Interfaces (IUF). , Miami, USA (2003).
13. Narayanan, A., Shmatikov, V.: Robust De-anonymization of Large Sparse Dataset. IEEE Symposium on Security and Privacy. , Oakland, USA (2008).
14. Soghoian, C.: AOL, Netflix and the end of open access to research data, http://www.cnet.com/news/aol-netflix-and-the-end-of-open-access-to-research-data/.
15. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. UMUAI. 22, 441–504 (2012).
16. Liu, B., Zhang, L.: A Survey of Opinion Mining and Sentiment Analysis. Mining Text Data. pp. 415–463 (2012).
17. Harris, Z.: Distributional Structure. Word. 10, 146–162 (1954).
18. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Document retrieval systems. pp. 143–160 (1988).
19. Zhanga, W., Yoshidab, T., Tang, X.: A comparative study of TF*IDF, LSI and multi-words for text classification. Expert Syst. Appl. 38, 2758–2765 (2011).
20. Wold, S.: Pattern recognition by means of disjoint principal components models. Pattern Recognit. 8, 127–139 (1976).
21. Dumais, S.T.: Latent semantic analysis. Annu. Rev. Inf. Sci. Technol. 38, 188–230 (2004).
22. Mnih, A., Hinton, G.E.: A scalable hierarchical distributed language model. Adv. Neural Inf. Process. Syst. 1081–1088 (2009).
23. Mikolov, T., Yih, W., Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language (NAACL-HLT). , Atlanta, USA (2013).
24. Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Empirical Methods in Natural Language Processing (EMNLP). , Doha, Qatar (2014).
25. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. Conf of the Neural Information Processing Foundation (NIPS). , Quebec, Canada (2014).
26. Hill, F., Cho, K., Jean, S., Devin, C., Bengio, Y.: Not All Neural Embeddings are Born Equal. NIPS Workshop on Learning Semantics. , Montreal, Canada (2014).
27. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. Conf. on Empirical Methods on Natural Language Processing (EMNLP). , Doha, Qatar (2014).
28. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. Int. Conf. on Machine Learning (ICML). pp. 1188–1196 (2014).
29. McAuley, J., Targett, C., Shi, J., Hengel, A. van den: Image-based recommendations on styles and substitutes. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR). , Santiago de Chile, Chile (2015).
30. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Mach. Learn. Res. 3, 993–1022 (2003).
31. Sen, S., Harper, F.M., LaPitz, A., Riedl, J.: The quest for quality tags. ACM Conf. on Supporting Group Work. , Sanibel Island, Florida, USA (2007).