

Exploiting Distributional Semantics Models for Natural Language Context-aware Justifications for Recommender Systems

Cataldo Musto
University of Bari, Italy
cataldo.musto@uniba.it

Giuseppe Spillo
University of Bari, Italy
giuseppe.spillo@studenti.uniba.it

Marco de Gemmis
University of Bari, Italy
marco.degemmis@uniba.it

Pasquale Lops
University of Bari, Italy
pasquale.lops@uniba.it

Giovanni Semeraro
University of Bari, Italy
giovanni.semeraro@uniba.it

ABSTRACT

In this paper we present a methodology to generate *context-aware natural language justifications* supporting the suggestions produced by a recommendation algorithm. Our approach relies on a natural language processing pipeline that exploits *distributional semantics models* to identify the most relevant *aspects* for each different context of consumption of the item. Next, these aspects are used to identify the most suitable pieces of information to be combined in a natural language justification. As information source, we used a corpus of reviews. Accordingly, our justifications are based on a combination of *reviews' excerpts* that discuss the aspects that are particularly relevant for a certain context.

In the experimental evaluation, we carried out a user study in the *movies* domain in order to investigate the validity of the idea of adapting the justifications to the different contexts of usage. As shown by the results, all these claims were supported by the data we collected.

Author Keywords

Recommender Systems, Explanation, Natural Language Processing, Opinion Mining

INTRODUCTION

Recommender Systems (RSs) [19] are now recognised as a very effective mean to support the users in decision-making tasks [20]. However, as the importance of such technology in our everyday lives grows, it is fundamental that these algorithms support each suggestion through a *justification* that allows the user to understand the *internal mechanisms* of the recommendation process and to more easily discern among the available alternatives.

To this end, several attempts have been recently devoted to investigate *how* to introduce explanation facilities in RSs [16] and to identify the most suitable explanation styles [4]. Despite such a huge research effort, none of the methodologies currently presented in literature diversifies the justifications based on the different *contextual situations* in which the item will be consumed. This is a clear issue, since context plays a key role in every decision-making task, and RSs are no exception. Indeed, as the *mood* or the company (*friends, family, children*) can direct the choice of the movie to be watched, so a justification that aims to convince a user to enjoy a recommendation should contain different concepts depending on whether the user is planning to watch a movie with her *friends* or with her *children*.

In this paper we fill in this gap by proposing an approach to generate a *context-aware* justification that supports a recommendation. Our methodology exploits distributional semantics models [5] to build a *term-context* matrix that encodes the importance of terms and concepts in each context of consumption. Such a matrix is used to obtain a vector space representation of each context, which is in turn used to identify the most suitable pieces of information to be combined in a justification. As information source, we used a corpus of reviews. Accordingly, our justifications are based on a combination of *reviews' excerpts* that discuss with a *positive* sentiment the aspects that are particularly relevant for a certain context. Beyond its *context-aware nature*, another distinctive trait of our methodology is the fact that we generate *post-hoc justifications* that are completely independent from the underlying recommendation models and completely separated from the step of generating the recommendations.

To sum up, we can summarize the contributions of the article as follows: (i) we propose a methodology based on *distributional semantics models* and *natural language processing* to automatically learn a vector space representation of the different *contexts* in which an item can be consumed; (ii) We design a pipeline that exploits *distributional semantics models* to generate *context-aware natural language justifications* supporting the suggestions returned by any recommendation algorithm;

The rest of the paper is organized as follows: first, in Section 2 we provide an overview of related work. Next, Section 3 describes the main components of our workflow and Section 4 discusses the outcomes of the experimental evaluation. Finally, conclusions and future work of the current research are provided in Section 5.

RELATED WORK

The current research borrows concepts from *review-based explanation strategies* and *distributional semantics models*. In the following, we will try to discuss relevant related work and to emphasize the hallmarks of our methodology.

Review-based Explanations. According to the taxonomy discussed in [3], our approach can be classified as a *content-based explanation strategy*, since the justifications we generate are based on descriptive features of the item. Early attempts in the area rely on the exploitation of tags [24] and features gathered from knowledge graphs [11]. With respect to classic content-based strategies, the novelty of the current work lies in the use of *review data* to build a natural language justification. In this research line, [2] Chen et al. analyze users’ reviews to identify relevant features of the items, which are presented on an *explanation interface*. Differently from this work, we did not bound on a fixed set of static aspects and we left the explanation algorithm deciding and identifying the most relevant concepts and aspects for each contextual setting. A similar attempt was also proposed in [1]. Moreover, as previously emphasized, a trait that distinguishes our approach with respect to such literature is the adaptation of the justification based on the different setting in which the item is consumed. The only work exploiting *context* in the justification process has been proposed by Misztal et al. in [9]. However, differently from our work, they did not diversify the justifications of *the same items* on varying of different contextual settings in which the item is consumed, since they just adopt features inspired by context (e.g., “*I suggest you this movie since you like this genre in rainy days*”) to explain a recommendation.

Distributional Semantics Models. Another distinctive trait of the current work is the adoption of *distributional semantics models* (DSMs) to build a vector space representation of the different *contextual situations* in which an item can be consumed. Typically, DSMs rely on a *term-context matrix*, where rows represent the terms in the corpus and columns represents *contexts of usage*. For the sake of simplicity, we can imagine a context as a fragment of text in which the term appears, as a sentence, a paragraph or a document. Every time a particular term is used in a particular context, such an information is encoded in this matrix. One of the advantages that follows the adoption of DSMs is that they can learn a vector space representation of terms in a totally *unsupervised* way. These methods, recently inspired methods in the area of *word embeddings*, such as WORD2VEC [8] and contextual word representations [21]. Even if some attempts evaluating RSs based on DSMs already exists [13, 12, 14], in our attempt we used DSMs to build a *vector-space representation of the different contextual dimensions*. Up to our knowledge, the usage of DSMs for justification purposes this is a completely new research direction in the area of explanation.

METHODOLOGY

Our workflow to generate context-aware justifications based on users’ reviews is shown in Figure 1. In the following, we will describe all the modules that compose the workflow.

Context Learner. The first step is carried out by the CONTEXT LEARNER module, which exploits DSMs to learn a vector space representation of the *contexts*. Formally, given a set reviews R and a set of k contextual settings $C = \{c_1 \dots c_k\}$, this module generates as output a matrix $C_{n,k}$ that encodes the importance of each term t_i in each contextual setting c_j . In order to build such a representation, we first split all the reviews $r \in R$ in sentences. Next, let S be the set of previously obtained sentences, we *manually annotated* a subset of these sentences in order to obtain a set $S' = \{s_1 \dots s_m\}$, where each s_i is labeled with one or more contextual settings, based on the concepts mentioned in the review. Of course, each s_i can be annotated with more than one context. As an example, a review including the sentence ‘*a very romantic movie*’ is annotated with the contexts *company=partner*, while the sentence ‘*perfect for a night at home*’ is annotated with the contexts *day=weekday*. After the annotation step, a *sentence-context* matrix $A_{m,k}$ is built, where each a_{s_i,c_j} is equal to 1 if the sentence s_i is annotated with the context c_j (that is to say, it mentions concepts that are relevant for that context), 0 otherwise.

Next, we run *tokenization* and *lemmatization* algorithms [7] over the sentences in S to obtain a *lemma-sentence* matrix $V_{n,m}$. In this case, v_{t_i,s_j} is equal to the TF/IDF of the term t_i in the sentence s_j . Of course, IDF is calculated over all the annotated sentences. In order to filter out non-relevant lemmas, we maintained in the matrix V just *nouns* and *adjectives*. Nouns were chosen due to previous research [15], which showed that descriptive features of an item are usually represented using nouns (e.g., service, meal, location, etc.). Similarly, adjectives were included since they play a key role in the task of catching the characteristics of the different contextual situations (e.g., romantic, quick, etc.). Moreover, we also decided to take into account and extract combinations of nouns and adjectives (bigrams) such as *romantic location*, since they can be very useful to highlight specific characteristics of the item.

In the last step of the process annotation matrix $A_{n,k}$ and vocabulary matrix $V_{m,n}$ are multiplied to obtain our *lemma-context* matrix $C_{n,k}$, which represents the final output returned by the CONTEXT LEARNER module. Of course, each $c_{i,j}$ encodes the importance of term t_i in the context c_j . The whole process carried out by this component is described in Figure 2.

Given such a representation, two different outputs are obtained. First, we can directly extract *column vectors* \vec{c}_j from matrix C , which represents the vector space representation of the context c_j based on DSMs. It should be pointed out that such a representation perfectly fits the principles of DSMs since contexts discussed through the same lemmas will share a very similar vector space representation. Conversely, a poor overlap will result in very different vectors. Moreover, for each column, lemmas may be ranked and those having the highest TF-IDF scores may be extracted. In this way, we obtain a *lexicon* of lemmas that are relevant for a particular contextual setting, and this can be useful to empirically validate the effectiveness

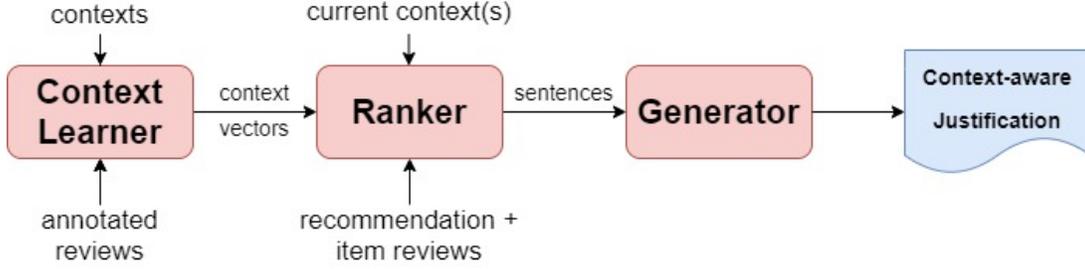


Figure 1: Workflow to generate Context-aware Justifications by Exploiting DSMs

$$\begin{pmatrix} v_{1,1} & v_{1,2} & \dots & v_{1,m} \\ v_{2,1} & v_{2,2} & \dots & v_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n,1} & v_{n,2} & \dots & v_{n,m} \end{pmatrix}_{V_{n,m}} \times \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,k} \\ a_{2,1} & a_{2,2} & \dots & a_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,k} \end{pmatrix}_{A_{m,k}} = \begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,k} \\ c_{2,1} & c_{2,2} & \dots & c_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,k} \end{pmatrix}_{C_{n,k}}$$

Figure 2: Building a *lemma-context* matrix C by exploiting distributional semantics models

of the approach. In Table 1, we anticipate some details of our experimental session and we report the top-3 lemmas for two different contextual settings starting from a set of *movie* reviews.

Ranker. Given a recommended item (along with its reviews) and given the context in which the item will be consumed (from now on, defined as ‘*current context*’), this module has to identify the most relevant *review excerpts* to be included in the justification. To this end, we designed a ranking strategy that exploits DSMs and similarity measures in vector spaces to identify suitable excerpts: given a set of n reviews discussing the item i , $R_i = \{r_{i,1} \dots r_{i,n}\}$, we first split each r_i in sentences. Next, we processed the sentences through a sentiment analysis algorithm [6, 17] in order to filter out those expressing a *negative* or *neutral* opinions about the item. The choice is justified by our focus on review excerpts discussing *positive* characteristics of the item. Next, let c_j be the current contextual situation (e.g., *company=partner*), we calculate the *cosine similarity* between the context vector \vec{c}_j returned by the CONTEXT LEARNER and a vector space representation of each sentence \vec{s}_i . The sentences having the highest cosine similarity w.r.t. to the context of usage c_j are selected as the most suitable excerpts and are passed to the GENERATOR.

Generator. Finally, the goal of GENERATOR is to put together the compliant excerpts in a single *natural language justification*. In particular, we defined a *slot-filling* strategy based on the principles of Natural Language Generation [18]. Such a strategy is based on the combination of a *fixed* part, which is common to all the justifications, and a *dynamic* part that depends on the outputs returned by the previous steps. In our case, the top-1 sentence for *each current* contextual dimension is selected, and the different excerpts are merged by exploiting simple connectives, such as adverbs and conjunctions. An example of the resulting justifications is provided in Table 2.

EXPERIMENTAL EVALUATION

The experimental evaluation was designed to identify the best-performing configuration of our strategy, on varying of different combinations of the parameters of the workflow (*Research Question 1*), and to assess how our approach performs in comparison to other methods (both context-aware and non-contextual) to generate *post-hoc justifications* (*Research Question 2*). To this end, we designed a user study involving 273 subjects (male=50%, degree or PhD=26.04%, age \geq 35=49.48%, already used a RS=85.4%) in the *movies* domain. Interest in *movies* was indicated as *medium* or *high* by 62.78% of the sample. Our sample was obtained through the *availability sampling* strategy, and it includes students, researchers in the area and people not skilled with computer science and recommender systems.

Experimental Design. To run the experiment, we deployed a web application¹ implementing the methodology described in Section 3. Next, as a first step, we identified the relevant *contextual dimensions* for each domain. Contexts were selected by carrying out an analysis of related work of context-aware recommender systems in the MOVIE domain. In total, we defined 3 contextual dimensions, that is to say, *mood* (*great, normal*), *company* (*family, friends, partner*) and *level of attention* (*high, low*). To collect the data necessary to feed our web application, we selected a subset of 300 popular movies (according to IMDB data) discussed in more than 50 reviews in the Amazon Reviews dataset². This choice is motivated by our need of a large set of sentences discussing the item in each contextual setting. These data were processed by exploiting lemmatization, POS-tagging and sentiment analysis algorithms available in CoreNLP³ and Stanford Sentiment Analysis algorithm⁴.

¹<http://193.204.187.192:8080/filmando-eng>

²<http://jmcauley.ucsd.edu/data/amazon/links.html> - Only the reviews available in the ‘Movies and TV’ category were downloaded.

³<https://stanfordnlp.github.io/CoreNLP/>

⁴<https://nlp.stanford.edu/sentiment/>

	<i>Attention=high</i>	<i>Attention=low</i>
Unigrams	engaging, attentive, intense	simple, smooth, easy
Bigrams	intense plot, slow movie, life metaphor	easy vision, simple movie, simple plot

Table 1: Top-3 lemmas returned by the CONTEXT LEARNER module for two couples of different contextual settings in the MOVIE domain.

Restaurant	Justification
Company=Partner	You should watch 'Stranger than Fiction'. It is a good movie to watch with your partner because <i>it has a very romantic end. Moreover, plot is very intense.</i>
Company=Friends	You should watch 'Stranger than Fiction'. It is a good movie to watch with friends since <i>the film crackles with laughter and pathos and it is a classy sweet and funny movie.</i>

Table 2: Context-aware justifications for the RESTAURANT domain. Automatically extracted review excerpts are reported in *italics*.

tool. Some statistics about the final dataset are provided in Table 3.

In order to compare different configurations of the workflow, we designed several variant obtained by varying the *vocabulary* of lemmas. In particular, we compared the effectiveness of simple *unigrams*, of *bigrams* and their merge. In the first case, we encoded in our matrix just single lemmas (*e.g.*, service, meal, romantic, etc.) while in the second we stored combination of nouns and adjectives (*e.g.*, romantic location). Due to space reasons, we can't provide more details about the lexicons we learnt, and we suggest to refer again to Table 1 for a qualitative evaluation of some of the resulting representations. Our representations based on DSMs were obtained by starting from a set of 1,905 annotations for the *movie* domain, annotated by three annotators by adopting a *majority vote* strategy. To conclude, each user involved in the experiment carried out the following steps:

1. *Training, Context Selection and Generation of the Recommendation.* First, we asked the users to provide some basic demographic data and to indicate their interest in *movies*. Next, each user indicated the *context of consumption* of the recommendation, by selecting a context among the different contextual settings we previously indicated (see Figure 3-a). Given the current context, a suitable recommendation was identified and presented to the user. As recommendation algorithm we used a content-based recommendation strategy exploiting users' reviews.
2. *Generation of the Justification.* Given the recommendation and the current context of consumption, we run our pipeline to generate a context-aware justification of the item *adapted to that context*. In this case, we designed a *between-subject* protocol. In particular, each user was randomly assigned to one of the *three* configurations of our pipeline and the output was presented to the user along with the recommendation (see Figure 3-b). Clearly, the user was not aware of the specific configuration he was interacting with.
3. *Evaluation through Questionnaires.* Once the justification was shown, we asked the users to fill in a post-usage questionnaire. Each user was asked to evaluate *transparency*,

persuasiveness, *engagement* and *trust* of the recommendation process through a five-point scale (1=strongly disagree, 5=strongly agree). The questions the users had to answer follow those proposed in [23]. Due to space reasons, we can't report the questions and we suggest to interact with the web application to fill in the missing details.

4. *Comparison to baselines.* Finally, we compared our method to two different baselines in a *within-subject* experiment. In this case, all the users were provided with two different justifications styles (*i.e.*, our context-aware justifications and a baseline) and we asked the users to choose the one they preferred. As for the baselines, we focused on other methodologies to generate post-hoc justifications and we selected (i) a context-aware strategy to generate justifications, which is based on a set of *manually defined relevant terms* for each context; (ii) a method to generate *non-contextual* review-based justifications that relies on the automatic identification of relevant aspects and on the selection of compliant reviews excerpts containing such terms. Such approach partially replicates that presented in [10].

Discussions of the Results Results of the first experiment, that allows to answer to *Research Question 1*, are presented in Table 4. The values in the tables represent the average scores provided by the users for each of the previously mentioned questions. As for the *movie* domain, results show that the overall best results are obtained by using a vocabulary based on *unigrams* and *bigrams*. This first finding provides us with an interesting outcome, since most of the strategies to generate explanations are currently based on *single keywords and aspects*. Conversely, our experiment showed that both *adjectives* as well as couples of co-occurring terms are worth to be encoded, since they catch more fine-grained characteristics of the item that are relevant in a particular contextual setting. Overall, the results we obtained confirmed the validity of the approach. Beyond the increase in TRANSPARENCY, high evaluations were also noted for PERSUASION and ENGAGEMENT metrics. This outcome confirms how the identification of relevant reviews' excerpts can lead to satisfying justifications. Indeed, differently from feature-based justifications, that typically rely on very popular and well-known characteristics of the movie, as the *actors* or the *director*, more specific aspects of the items

	#Items	#Reviews	#Sentences	#Positive Sent.	Avg. Sent./Item	Avg. Pos. Sent./Item
MOVIES	307	153,398	1,464,593	560,817	4,770.66	1,826.76

Table 3: Statistics of the dataset

(a) Context Selection

(b) Recommendation and Justification

Figure 3: Interaction with the web application.

emerge from users' reviews.

Next, in order to answer to **Research Question 2**, we compared the best-performing configurations emerging from Experiment 1 to two different baselines. The results of these experiments are reported in Table 5 which show the percentage of users who preferred our context-aware methodology based on DSMs to both the baselines. In particular, the first comparison allowed us to assess the effectiveness of a *vector space* representation of contexts based on DSMs with respect to a simple context-aware justification method based on a *fixed* lexicon of relevant terms, while the second comparison investigated how valid was the idea of diversifying the justifications based on the different contextual settings in which the items is consumed. As shown in the table, our approach was the preferred one in both the comparisons. It should be pointed out that the gaps are particularly large when our methodology is compared to a *non-contextual* baseline. In this case, we noted a statistically significant gap ($p \leq 0.05$) for all the metrics, with the exception of *trust*. This suggests that diversifying the justifications based on the *context of consumption* is particularly appreciated by the users. This confirms the validity of our intuition, which led to a completely new research direction in the area of justifications for recommender systems.

CONCLUSIONS AND FUTURE WORK

In this paper we presented a methodology that exploits DSMs to build post-hoc *context-aware* natural language justifications supporting the suggestions generated by a RS. The hallmark of this work is the diversification of the justifications based on the different contextual settings in which the items will be consumed, which is a new research direction in the area. As shown in our experiments, our justifications were largely

preferred by users. This confirms the effectiveness of our approach and paves the way to several future research directions, such as the definition of *personalized* justifications as well as the generation of *hybrid* justifications that combine elements gathered from user-generated content (as the reviews) with descriptive characteristics of the items. Finally, we will also evaluate to what extent these justifications can *explain* the behavior of complex and non-scrutable models such as those based on complex deep learning techniques [22].

REFERENCES

- [1] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-based Personalized Natural Language Explanations for Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 175–182.
- [2] Li Chen and Feng Wang. 2017. Explaining Recommendations based on Feature Sentiments in Product Reviews. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*. ACM, 17–28.
- [3] Gerhard Friedrich and Markus Zanker. 2011. A taxonomy for generating explanations in recommender systems. *AI Magazine* 32, 3 (2011), 90–98.
- [4] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies* 72, 4 (2014), 367–382.
- [5] Alessandro Lenci. 2008. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics* 20, 1 (2008), 1–31.

<i>Metrics / Configuration</i>	<i>Unigrams</i>	<i>Bigrams</i>	<i>Uni+Bigrams</i>
<i>Transparency</i>	3.38	3.81	3.64
<i>Persuasion</i>	3.56	3.62	3.54
<i>Engagement</i>	3.54	3.72	3.70
<i>Trust</i>	3.44	3.66	3.61

Table 4: Results of Experiment 1 for the MOVIE domain. The best-performing configuration is reported in **bold** and underlined

<i>Metrics / Choice</i>	vs. Context-aware Static Baseline			vs. Non-Contextual Baseline		
	<i>CA+DSMs</i>	<i>Baseline</i>	<i>Indifferent</i>	<i>CA+DSMs</i>	<i>Baseline</i>	<i>Indifferent</i>
<i>Transparency</i>	52.38%	38.10%	19.52%	53.21%	34.47%	12.32%
<i>Persuasion</i>	54.10%	36.33%	19.57%	55.17%	32.33%	12.50%
<i>Engagement</i>	49.31%	39.23%	11.56%	44.51%	32.75%	22.74%
<i>Trust</i>	42.86%	39.31%	17.83%	42.90%	42.11%	14.99%

Table 5: Results of Experiment 2, comparing our approach (*CA+DSMs*) to a context-aware baseline that does not exploit DSMs (*CA Static*) and to a non-contextual baseline that exploit users' reviews (*review-based*). The configuration preferred by the higher percentage of users is reported in **bold**.

- [6] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.
- [7] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.
- [8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
- [9] Joanna Misztal and Bipin Indurkha. 2015. Explaining Contextual Recommendations: Interaction Design Study and Prototype Implementation.. In *IntRS@ RecSys*. 13–20.
- [10] Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2020. Generating post hoc review-based natural language justifications for recommender systems. *User Modeling and User-Adapted Interaction* (2020), 1–45.
- [11] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2016. ExpLOD: A Framework for Explaining Recommendations Based on the Linked Open Data Cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 151–154. DOI: <http://dx.doi.org/10.1145/2959100.2959173>
- [12] C. Musto, F. Narducci, P. Lops, G. Semeraro, M. De Gemmis, M. Barbieri, J. Korst, V. Pronk, and R. Clout. 2012. Enhanced semantic TV-show representation for personalized electronic program guides. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7379 LNCS (2012), 188–199. DOI: http://dx.doi.org/10.1007/978-3-642-31454-4_16 cited By 19.
- [13] C. Musto, G. Semeraro, P. Lops, and M. de Gemmis. 2011. Random Indexing and Negative User Preferences for Enhancing Content-Based Recommender Systems. In *EC-Web 2011 (Lecture Notes in Business Inf. Processing)*, Vol. 85. Springer, 270–281.
- [14] C. Musto, G. Semeraro, P. Lops, and M. de Gemmis. 2014. Combining distributional semantics and entity linking for context-aware content-based recommendation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8538 (2014), 381–392. DOI: http://dx.doi.org/10.1007/978-3-319-08786-3_34 cited By 19.
- [15] Hiroshi Nakagawa and Tatsunori Mori. 2002. A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology-Volume 14*. Association for Computational Linguistics, 1–7.
- [16] Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* 27, 3-5 (2017), 393–444.
- [17] Gerald Petz, Michał Karpowicz, Harald Fürschuß, Andreas Auinger, Václav Střiteský, and Andreas Holzinger. 2015. Reprint of: Computational approaches for mining user's opinions on the Web 2.0. *Information Processing & Management* 51, 4 (2015), 510–519.
- [18] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3, 1 (1997), 57–87.
- [19] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

- [20] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender systems handbook*. Springer, 1–34.
- [21] Noah A Smith. 2020. Contextual word representations: putting words into computers. *Commun. ACM* 63, 6 (2020), 66–74.
- [22] A. Suglia, C. Greco, C. Musto, M. De Gemmis, P. Lops, and G. Semeraro. 2017. A deep architecture for content-based recommendations exploiting recurrent neural networks. *UMAP 2017 - Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization* (2017), 202–211. DOI: <http://dx.doi.org/10.1145/3079628.3079684> cited By 20.
- [23] Nava Tintarev and Judith Masthoff. 2012. Evaluating the Effectiveness of Explanations for Recommender Systems. *UMUAI* 22, 4-5 (2012), 399–439.
- [24] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 47–56.