User Modeling and User-Adapted Interaction
https://doi.org/10.1007/s11257-023-09382-x

Harnessing distributional semantics to build context-aware justifications for recommender systems

Cataldo Musto[1] · **Giuseppe Spillo**[1] · **Giovanni Semeraro**[1]

/ Accepted in revised form: 17 August 2023

Abstract
This paper introduces a methodology to generate *review-based* natural language justifications supporting personalized suggestions returned by a recommender system. The hallmark of our strategy lies in the fact that natural language justifications are adapted to the different *contextual situations* in which the items will be consumed. In particular, our strategy relies on the following intuition: Just like the selection of the most suitable item is influenced by the contexts of usage, a justification that supports a recommendation should vary as well. As an example, depending on whether a person is going out with her friends or her family, a justification that supports a *restaurant recommendation* should include different *concepts* and *aspects*. Accordingly, we designed a pipeline based on *distributional semantics models* to generate a vector space representation of each context. Such a representation, which relies on a *term-context* matrix, is used to identify the most suitable review excerpts that discuss aspects that are particularly relevant for a certain context. The methodology was validated by means of two user studies, carried out in two different domains (i.e., movies and restaurants). Moreover, we also analyzed whether and how our justifications impact on the perceived *transparency of* the recommendation process and allow the user to make more *informed* choices. As shown by the results, our intuitions were supported by the user studies.

Keywords Recommender systems · Natural language processing · Opinion mining · Dialog · Preference elicitation · Virtual assistants

✉ Cataldo Musto
cataldo.musto@uniba.it

Giuseppe Spillo
g.spillo@studenti.uniba.it

Giovanni Semeraro
giovanni.semeraro@uniba.it

[1] Dipartimento di Informatica, University of Bari Aldo Moro, Piazza Umberto I 1, Bari 70125, Italy

Published online: 25 September 2023

1 Introduction

About thirty years after the birth of the first Recommender Systems (RSs) (Resnick and Varian 1997), this technology is now recognized as a very effective means to tackle the problem of *information overload* and to support the users in decision-making tasks (Shapira et al. 2022). The effectiveness of these algorithms, originally developed for low-risk decision-making domains (music to listen to, or movie to watch), is nowadays validated by its recent spread in more complex and high-risk scenarios, such as health, finance and so on Jameson et al. (2022).

However, in parallel with the spread of these technologies in our everyday lives, it is more and more required that RSs *open their black boxes* (Guidotti et al. 2019). In other terms, they have to: (i) make their *internal mechanisms* as much clear and transparent as possible; (ii) support each suggestion by means of *justifications* that allow the user to more easily discern among the available alternatives. As shown by several research, a higher *transparency* of the whole recommendation process leads to a higher *trust* of the users (Sinha and Swearingen 2002) as well as to a higher *acceptance* of the recommendations (Cramer et al. 2008).

Accordingly, the introduction of explanation facilities in RSs has been investigated in several works (Nunes and Jannach 2017). Generally speaking, such explanations range from attempts based on the descriptive properties of the items (Vig et al. 2009) to strategies based on deep learning that jointly learn to recommend and explain (Lu et al. 2018; Liu et al. 2019). Recently, some *black box* models that exploit knowledge graphs to generate *post hoc explanations* that are independent from the recommendation algorithm (Musto et al. 2016) also emerged.

Regardless of the richness and the effectiveness of the explanations these strategies can build, the intuition of adapting the explanations to the different *contextual conditions* in which the item is consumed has been scarcely investigated. This is a relevant concern, since the context plays a fundamental role when a decision (e.g., a movie to watch) shall be made, and RSs are no exception. As an example, the *mood* can direct the choice of the movie to be watched, and the company (*friends, family, children*) can drive the choice of a restaurant. In the same way, a justification supporting a restaurant recommendation should convey different concepts depending on whether the user is arranging a *family lunch* or a *dinner with friends*.

In this paper, we present a novel strategy to effectively tackle this issue since we introduce an approach to build *review-based context-aware justifications* supporting the recommendation. Our methodology is inspired by the principles of distributional semantics (Lenci 2008) and is based on three steps: First, a *term-context* matrix that encodes the importance of terms and concepts in each contextual condition is built. Thanks to this matrix, it is possible to obtain a vector space representation of each context, which is used to identify the most suitable pieces of information (e.g., review excerpts) to be included in the justification. Generally speaking, our justifications rely on the most relevant *reviews' excerpts* that discuss with a *positive* sentiment the aspects that are relevant in a particular combination of contextual conditions. The choice of preferring *reviews' excerpts* w.r.t. other descriptive features is due to two main reasons: First, review excerpts often convey information about what people *like* and what people *think* about the places they visit and the experiences they have, and

this can be very helpful to discover new aspects of the items or to be persuaded to consume an item. Moreover, *review-based explanations* tend to overcome those based on descriptive features (He et al. 2015; Baral et al. 2018). In particular, previous works showed that the use of review data improves the engagement and the persuasive power of the explanations. This is probably due to the different nature of features included in the justifications that allow the users to discover new things about the item.

Another distinctive trait of our strategy is the generation of *post hoc justifications*. These justifications are completely independent from the algorithm which is used to generate the recommendations. Accordingly, our methodology works in a completely *unsupervised* way and does not need *users' ratings* as input since our strategy needs to be just fed with: (i) a recommended item; (ii) a set of reviews discussing the item; (iii) the context in which the items will be consumed and returns as output a suitable context-aware justification. However, it is important to point out that the current implementation of the model returns justifications are not *personalized*, that is to say, given a certain contextual condition, two users who receive the same recommendation will receive the same justification as well. More details about this will be provided later.

Finally, it is also necessary to emphasize that, throughout the article, the term *justification* is preferred to *explanation*. This choice is mainly due to the principles introduced in Biran and Cotton (2017), where the authors state that explanations are related to the concept of interpretability, that is to say, *if the operations of the algorithm can be understood by a human*. Conversely, *justifications explain why a decision is a good one*. The latter definition perfectly frames the ideas discussed in this article, since our justifications do not consider the internal operations the recommendation algorithm carries out and are more oriented to describe *why* a user would be interested in the item. To sum up, our article provides the following contributions.

- We introduce a strategy inspired by *distributional semantics models* that exploits *natural language processing* to learn a representation of the different *contextual conditions* in which an item can be consumed;
- We design a pipeline to generate *review-based natural language justifications* adapted to the different contextual situation;
- We carry two user studies in the movies and restaurants domain to assess the effectiveness of the methodology.

The rest of the paper is organized as follows: Sect. 2 provides an overview of related work. Next, the main components of our workflow are discussed in Sect. 3. Section 4 presents the experimental settings and discusses its outcomes. Finally, Sect. 6 summarizes conclusions and future work of the current research.

2 Related work

This work borrows concepts from *context-aware computing*, *review-based explanation strategies* and *distributional semantics models*. Accordingly, in this section we introduce relevant related work in these areas, and we try to indicate the distinctive traits of our methodology.

2.1 Context-aware computing

According to the literature in the area of context-aware computing (Schilit et al. 1994), the context can be defined as a *set of factors that describe the current situation and can potentially influence the decision-making process*. As for the area of RSs, the positive impact of contextual information is acknowledged from around one decade (Adomavicius et al. 2022) since most of the current literature showed that context-aware recommender systems (CARS) usually outperform their non-contextual-aware version (Adomavicius et al. 2005).

As regarding the terminology we will use throughout the article, we want to emphasize we used the definitions that are commonly used in the literature: From now on, we use the term *contextual variable* to refer to a variable that describes the current situation (e.g., mood, company), the term *contextual condition* to refer to a value that a contextual variable can assume (e.g., mood=good), while a *contextual situation* is a combination of contextual conditions (e.g., mood=good, company=friends).

The introduction of explanation strategies specifically designed for CARS is more recent. As an example, in Mei et al. (2018) the authors develop a neural context-aware recommender system that aims to generate *explainable* recommendations. In particular, explanation mechanisms are based on the analysis of the *attention weights* that are spread in the neural network. A similar attempt was also proposed in Xia et al. (2017). Our work significantly differs from these attempts since we developed a *black box* methodology which is separate from the generation of the recommendations. Our methodology generates *post hoc* justifications that are completely independent of the underlying recommendation model, so we are not interested in *how* recommendations are generated.

A complete discussion of the recent advances in the area of CARS (Haruna et al. 2017) is out of the scope of the current paper, since we do not investigate how *contextual conditions* influence the generation of the recommendations. Conversely, we adapt concepts and the aspects that appear in a *justification* supporting the recommendation based on the different contexts of usage. The only similar attempt is due to Misztal and Indurkhya (2015), who exploit contextual features to explain a recommendation (e.g., *"I suggest you this movie since you often like romantic movies in rainy days"*). A similar attempt was proposed by Sato et al. (2018), who identified the suitable contexts of consumption of an item, and uses them as a novel explanation style (e.g., *"This restaurant is recommended to you because it is suitable for dates with your partner."*). A similar explanation style is also presented by Li et al. (2021), who propose a method to jointly provide context-aware recommendations and explanations that exploit attention mechanisms.

With respect to these pieces of work, the distinctive trait of our methodology lies in the fact that we do not limit at indicating the contextual conditions in which an item is suitable. Conversely, we highlight the characteristics of the items that make it suitable in a particular contextual situation, and such characteristics are adapted as the setting changes.

2.2 Review-based explanations

Based on the classification of explanation algorithms presented in Friedrich and Zanker (2011), our methodology can be classified as a *content-based explanation strategy*. Indeed, we generate justifications by exploiting descriptive features of the item. Early attempts in the area rely on the exploitation of user-generated content, such as tags (Vig et al. 2009) or features gathered from knowledge sources and knowledge graphs (Musto et al. 2016). Differently from simple content-based strategies, the distinctive trait of our method lies in the use of *review data* to generate natural language justifications. Even if the positive impact of review data on recommendation quality is acknowledged in the literature (Hernández-Rubio and Bellogín 2018), the exploitation of these features to generate review-based and context-aware justifications is a relatively new research area.

Several work recently exploits review data to generate *explainable recommendations*. As an example, Baral et al. (2018) identify relevant aspects in the reviews by using deep neural networks and exploit a user-aspect bipartite graph to generate *explainable recommendations*. Similarly, He et al. (2015) encoded a user-item-aspect graph and exploited graph-based ranking to find the most relevant aspects of a place that match users' interests as well.

Finally, Chen et al. present in (2018) a model called NARRE that relies on a Neural Attentional Regression providing also review-level explanations. However, all these techniques were not taken into account for experiments since they all focus on the use or ratings and reviews to generate *explainable recommendations*. Conversely, our work presents a strategy to generate *post hoc* explanations that are independent from the underlying algorithm. Accordingly, in order to provide a fair comparison, similar methodologies proposing strategies for post hoc natural language explanations were considered as baselines.

The idea of analyzing users' reviews to identify relevant features of the item is also investigated in Chen and Wang (2017). However, Chen et al. only use in (2017) a *pre-defined set* of descriptive features. The distinctive trait of our work lies in the fact that we did not build our justifications by exploiting a fixed set of static aspects. Conversely, the most relevant concepts in a certain contextual condition are identified by our framework. Similarly, a framework to generate review-based explanations is also discussed in Muhammad et al. (2016). However, in this case the authors just identify relevant aspects of the items (e.g., bar, service, parking, etc.) without providing the user with a natural language explanation. Differently from this work, we combine excerpts of the original reviews to build a natural language justification. Similarly, Chang et al. (2016) investigated an approach based on the same principles. However, they strongly rely on crowd-sourcing, while our approach is based on a pipeline which

is fully automated, by excluding an initial effort to identify contextual conditions and to annotate review excerpts that are relevant to justify a recommendation. Moreover, the hallmark of our strategy w.r.t. all these work lies in the ability to differentiate the justification based on the different setting in which an item is consumed.

2.3 Distributional semantics models

Finally, it is worth to emphasize that this work exploits *distributional semantics* (DSMs) to build a content-based vector space representation of each *contextual condition* in which the item may be consumed. Generally speaking, DSMs have their roots in the *distributional hypothesis*, which states that '*Words that occur in the same contexts tend to have similar meanings*" (Harris 1968). DSMs are also referred to as *geometrical models*, since each term can be represented as a vector in geometrical space called WordSpace (Lowe 2001). All the methods that fall into this group share the same underlying principle: As humans infer the meaning of a word by analyzing the contexts in which the word is often used, so DSMs learn a representation of terms (i.e., its '*meaning*') by analyzing how the term itself is used in a corpus of textual documents.

The core of DSMs lies in the construction of a matrix called *term-context matrix*. This matrix encodes each term in a row, and each *contexts of usage* in a column. Generally speaking, a *context* can be figured out as a particular fragment of text in which the term appears. It can be a sentence, a paragraph or even a complete document. In a nutshell, a *term-context matrix* encodes how many times a particular term is used in a particular context, and each row in such a matrix is a *vector* which is instantiated in a vector space represented by the column of the term-context matrix.

Accordingly, the resulting representation of each word—that can be learnt in a completely unsupervised way—depends on the contexts in which the word itself appears (e.g., other words it co-occurs with), and this follows the principles of DSMs.

These methods, which took their roots in the area of *computational linguistics* (Lenci 2008), inspired recent and well-performing methods in the area of *word embeddings*. One of the examples is Word2Vec (Mikolov et al. 2013), followed by more recent contextual word representations (Smith 2020). However, the flexibility of DSMs makes them suitable for several heterogeneous tasks. As an example, in Musto et al. (2011) the authors propose a content-based RS that exploits DSMs. Similarly, Codina et al. (2016) use DSMs to learn a vector space representation of each contextual condition. However, differently from our work, in Codina et al. (2016) the authors exploit user ratings (rather than users' reviews, as we do) and use such a representation to calculate the similarity between contextual conditions and to alleviate *sparsity* problems in collaborative RSs. Conversely, in our attempt we used DSMs to build *a vector space representation of the different contextual conditions* which is used for justification purposes.

Finally, we want to point out that our work is partially inspired by Staiano and Guerini (2014), where Staiano et al. use DSMs to generate a vector space representation of *emotions*. In particular, they manually annotate sentences with the emotions they triggered and they build a *word-emotion* matrix, from which *emotion vectors* are

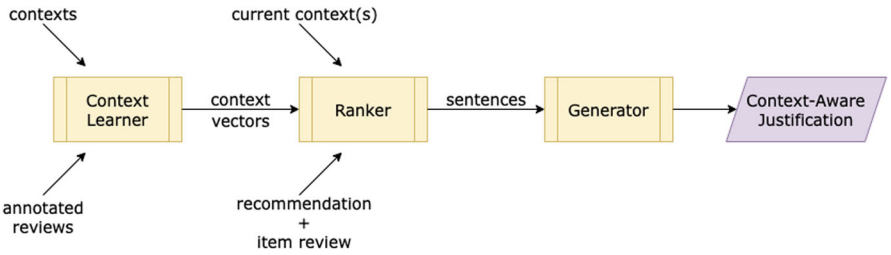

Fig. 1 Workflow to generate context-aware justifications by exploiting DSMs

extracted. Similarly, in our work we annotated reviews with the contextual conditions in which they can be useful, and we built a vector space representation of contexts which is then used to identify suitable review excerpts to combine in a justification. More details about this will be provided later. Based on our knowledge in the area, there is no attempt investigated how to exploit DSMs to generate justifications for RSs.

3 Methodology

In a nutshell, our methodology to generate natural language context-aware justifications takes as input: (i) a *recommendation*; (ii) a *set of reviews discussing the recommended item*; (iii) a *context of usage*. Based on this input, a *justification* that supports the recommendation is generated as output. As previously explained, justifications will combine several review excerpts and are adapted to the different contextual situations in which the item is consumed.

Figure 1 depicts the structure of our workflow to generate natural language review-based context-aware justifications. It consists of three main components: (i) a CONTEXT LEARNER that learns a representation of each contextual condition by exploiting DSMs; (ii) a RANKER implementing a scoring mechanism to identify the most suitable *review excerpts* that can support the recommendation; (iii) a GENERATOR that puts together previously retrieved pieces of information and provides the user with a context-aware justification of the item. In the following, we will provide more details for all the components.

Moreover, it is important to point out that Fig. 1 further clarifies the *post hoc* nature of our justifications since, in order to run our pipeline, it is just necessary to provide a *recommended item* and a set of reviews discussing that item, with the contextual information characterizing the user at the moment of the recommendation. This holds regardless of the algorithm which is used to generate the recommendation. In our opinion, this is a strong point of our methodology, since we are able to *justify* a suggestion even without a minimum amount of ratings, which is typically mandatory to learn a complex *explanation model* as those previously presented in the related work. Of course, in order to extend the findings of the evaluation, we will assess the effectiveness of the framework with different recommendation algorithms. This is left as future work.

3.1 Context learner

As we previously introduced, our strategy has the distinctive trait of adapting the content provided in the justification based on the contextual situations. Accordingly, as a first step, the system has to: (i) learn a *representation* for each contextual condition in which the item is consumed; (ii) to identify some review excerpts that emphasize and discuss features of the item that can be relevant in that specific contextual situation.

In our workflow, this task is carried out by the CONTEXT LEARNER. This component exploits DSMs and learns a space representation of each *context* based on relevant review excerpts. Intuitively, through this representation our system can identify *terms* and *concepts* that are relevant in a particular contextual condition.

Even if some work proposing strategies for the automatic identification of relevant contextual condition exists (Adomavicius et al. 2022), in this work the set of contextual variables that are relevant in a particular domain has to be manually defined. However, this is common to several applications that are context-aware in their nature and does not need to be carried out from scratch, since there is a large body of the literature providing some examples of relevant contextual variables. More details on the context definition process will be provided in Section 4.

Formally, given a set of reviews R, we first use natural language processing techniques to obtain a vocabulary of terms $T = \{t_1 \dots t_n\}$ occurring in the reviews. Next, given a set of contextual conditions $X = \{x_1 \dots x_k\}$ (e.g., company=family, meal=dinner, mood=good, etc.), this module generates as output a matrix $C_{n,k}$, where n is the size of the vocabulary T and k is the number of contextual conditions. In particular, each c_{t_i,x_j} in $C_{n,k}$ encodes how important a term t_i is in a particular contextual condition x_j.

In order to build such a representation, the following process was carried out: As a first step, all the reviews $r \in R$ are split into sentences. Next, let S be the set of previously obtained sentences, a subset of these sentences is *manually annotated* to obtain a set $S' = \{s_1 \dots s_m\}$. In this case, each s_i is labeled with one or more contextual conditions, based on the concepts that are used 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 funny and informal location*' is annotated with the contexts *company=friends* and *meal=dinner*, while the sentence '*many services for kids and families*' is annotated with the contexts *meal=lunch* and *company=family*.[1]

Based on our intuitions, the first group of annotations can be helpful to identify review excerpts assessing that a particular restaurant is suitable for a dinner with friends. Similarly, the second group of annotations highlights the excerpts that allow to select a restaurant for a family lunch. Once all the annotations are collected, it is possible to build a *sentence-context* matrix $A_{m,k}$, where m is the number of annotated sentences in S', while k is the number of contextual conditions, again. Each element a_{s_i,x_j} in the matrix is set as 1 if sentence s_i is annotated with the context x_j (that is to say, it mentions concepts that are relevant for the context), 0 otherwise.

[1] In order to get the annotations, the following question was formulated: '*In what contextual situation can such an excerpt be useful for justifying a recommendation?*.'

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

Fig. 2 Building a *lemma-context* matrix C by exploiting distributional semantics models

Next, each sentence is split into *terms*. This is done to identify the *concepts* that are relevant for a particular contextual condition. To get the list of the terms, *tokenization* and *lemmatization* algorithms (Manning et al. 1999) are run over the set of sentences in S, and a *lemma-sentence* matrix $V_{n,m}$ is obtained as output. In this case, v_{t_i,s_j} is equal to the TF- IDF of the term t_i in the sentence s_j.[2]

As a further step, in order to reduce the size of the vocabulary T, we filtered out non-useful lemmas. In this case, our design choice was to maintain *nouns* and *adjectives*. This choice put its root in previous research (Nakagawa and Mori 2002), where the authors showed that just nouns (e.g., service, meal, location, etc.) are typically used to label descriptive features of the items. Conversely, the choice of considering adjectives does not rely on related work, and it is due to the fact that adjectives are often used to model and describe the characteristics of the different contextual situations (e.g., romantic, quick, etc.).

Moreover, in order to introduce more significant concepts in our justifications, we also considered bigrams (i.e., couples of terms composed of nouns and adjectives, such as *elegant location*). Based on our intuition, bigrams can be very useful to highlight specific and context-aware characteristics of the item. To this end, given the set S' containing the annotated sentences, the POS-tagging algorithm (Manning 2011) was exploited and all the lemmas whose part of speech was equal to *nouns* and *adjectives*,[3] were maintained.

The *lemma-context* matrix $C_{n,k}$, which is the final output of the process, is finally obtained by multiplying the vocabulary matrix $V_{n,m}$ and the annotation matrix $A_{m,k}$. In $C_{n,k}$, each c_{t_i,x_j} encodes the importance (weight) of term t_i in the context x_j. The process is shown in Fig. 2, and this represents the final output generated by the CONTEXT LEARNER module.

Based on this representation, it is possible to obtain two different outputs. First, each column vector $\vec{c_j}$ can be extracted from matrix C. The vector space representation of the context x_j based on DSMs can be obtained by extracting each vector $\vec{c_j}$.

It is worth to emphasize that our column vectors perfectly fit with the principles of DSMs and *distributional hypothesis*. Indeed, whenever a particular context is labeled by the users by using similar (or same) lemmas, the resulting vector space representation will be similar as well. Conversely, different word usage will lead to a very different

[2] Of course, the calculation of IDF is repeated over all the annotated sentences.

[3] In particular, we maintained lemmas whose POS-tag was equal to NN NNS, NNP, NNPS, JJ, JJS and JJR that correspond to nouns, adjectives and their comparatives and superlatives. An overview of all the grammatical categories is out of the scope of this paper. However, based on the Penn Treebank tagset, which is one of the most popular tagsets, 36 categories for words can be defined. For further reading, we refer the reader to Marcus et al. (1994).

word representations. More evidence about this will be provided in the experimental section.

Moreover, a *lexicon* of lemmas that are relevant for a particular contextual condition can be easily obtained by further processing the matrix. As an example, for each column vector, lemmas could be ranked based on their descending TF- IDF score and those with the highest score could be labeled as most distinctive lemmas for that context. In order to provide some quantitative evidence of the effectiveness of the approach, Table 1 reports top 5 lemmas for two different contextual conditions for the *restaurant* and the *movie* domains, respectively. At first glance, the table confirms the effectiveness of our methodology, since top-ranked lemmas correctly catch and emphasize concepts that are particularly relevant for that specific context of usage.

As shown in table, very different aspects are highlighted for each context. Some lemmas may be common to different settings, but each contextual condition is characterized by a very specific vocabulary of terms that have to be used to justify the recommendation. As an example, sentences such as '*a funny location*' and '*many services for kids*' both describe positive aspects of the item; thus, it would make sense to include both of them in a justification.

However, the inclusion of a particular term in a justification is strictly dependent on the importance of the term itself in that specific context of usage. As an example, the first sentence can be used to convince that a restaurant is good for a dinner with friends, while the second can be exploited to drive a user toward a good restaurant for a family lunch.

3.2 Ranker

Once a vector space representation of the contexts is obtained, the RANKER component comes into play. In particular, the goal of this module is to identify the most relevant *review excerpts* to be included in the justification, given a recommended item, some reviews discussing the item, and the context of usage (from now on, defined as '*current context*').

Our ranking strategy is mainly inspired by DSMs and similarity measures for vector spaces, and works as follows: Given a set of n reviews discussing the item i, $R_i = \{r_{i,1} \dots r_{i,n}\}$, each r_i is first split into sentences. Next, by exploiting a sentiment analysis algorithm (Liu 2012) all the sentences are processed and those expressing a *negative* or *neutral* opinion are filtered out. The choice is due to the fact that we wanted to include in our justifications review excerpts discussing *positive* characteristics of the item.[4]

Next, in order to identify the most relevant sentences for the current contextual condition x_j (e.g., *company=partner*), we extract its vector space representation $\vec{c_j}$ and we calculate the *cosine similarity* between $\vec{c_j}$ and all the available sentences $\vec{s_i}$.

Given that *context vectors* are represented in an *n-dimensional* space where each element encodes the importance of the lemma in a particular context, to calculate such a ranking we need to build an *n-dimensional* representation for *sentence vectors*,

[4] A relevant research direction for future work could regard the introduction of *negative* aspects of the items, in order to evaluate their impact on the users.

Table 1 Output of the CONTEXT LEARNER for two couples of different contextual conditions

RESTAURANTS				MOVIES			
Company=partner		*Company=family*		*Attention=high*		*Attention=low*	
Unigrams	Bigrams	Unigrams	Bigrams	Unigrams	Bigrams	Unigrams	Bigrams
Romantic	Romantic evening	Kids	Large area	Engaging	Intense plot	Simple	Easy vision
Music	Warm Atmosphere	Playground	Kids room	Attentive	Slow film	Smooth	Simple movie
Terrace	Right Atmosphere	Clean	Excellent price	Intense	Life metaphor	Easy	Simple plot
Cozy	Warming Welcome	Family	Family welcome	Thriller	Engaging movie	Fun	Engaging end
Elegant	Live music	Children	Very airy	Victim	Mature topic	Clear	Young movie

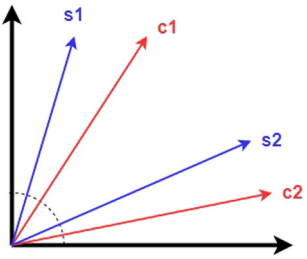

Fig. 3 Context-driven sentence selection carried out by the RANKER module. It is possible to observe that, for context *c1*, the sentence *s1* is the closest in the space and will probably be more characterizing than sentence *s2*; a similar observation can be made for context *c2* and sentence *s2*

as well. Accordingly, we first tokenize and lemmatize each sentence as we did in Sect. 3.1. Next, we instantiate the vector $\vec{s_i}$ in the same space defined by the *lemma-context* matrix $C_{n,k}$. In particular, $\vec{s_i} = [t_1, t_2 \ldots t_n]^\top$, where each t_j represents the TF- IDF score of the term. Clearly, TF counts how many times t_j appears in s_i, while IDF is calculated in the canonical way. Finally, the most similar sentences (according to cosine similarity) to the *vector* representing the context of usage x_j are selected as the most suitable sentences to be included in the justifications, and they are passed to the GENERATOR.

A *toy* example showing the behavior of the RANKER module is reported in Fig. 3. In particular, we instantiate two different *sentence vectors* s_1 = '*A very romantic location*' and s_2 = '*Includes a very nice playground for kids*' and two different context vectors. x_1 = '*company=partner*' and x_2 = '*company=family*.' As shown in Fig. 3, as expected the different contexts are far away from each other in the vector space, and a similar intuition also holds for the sentences. Indeed, s_1 is closer to x_1, while s_2 is closer to x_2. Accordingly, s_1 will result as the most suitable sentence to justify item i in context x_1, while s_2 will justify the recommendation if x_2 is the context of consumption. Thanks to this example, we have shown that two *different* justifications can be generated for the *same* recommendation, based on the different context in which the item will be consumed. As shown throughout this section, DSMs can be very helpful to achieve this goal.

Of course, this process works if just *one* contextual condition characterizes the current context of usage. However, it is frequent that several contextual conditions occur together (e.g., *company=friends* and *meal=lunch*). In this case, we propose two different strategies, which are labeled as *separate* and *centroid*, respectively. In the first case, we repeat the process for *each* current context x_j, and each execution will return a set of sentences that will be merged in the final justification. Conversely, in the latter we run the pipeline just once by first calculating the *centroid vector* $\vec{c}$ that merges the vectors $\vec{c_j}$, modeling the different current contextual situations. In the experimental evaluation, we will compare the effectiveness of these strategies to rank the available review excerpts.

3.3 Generator

When all the suitable and compliant review excerpts have been identified by the RANKER module, the GENERATOR is run to put everything together and build the *natural language justification* to be presented to the user. Our strategy for natural

language generation (NLG) (Reiter and Dale 1997) is based on template-based NLG (Deemter et al. 2005) and follows the principles of *slot-filling*. As stated in Deemter et al. (2005), these classes of approaches obtain performances that are often comparable to that obtained by other more sophisticated NLG techniques.

In our case, each justification is obtained by combining some *fixed* parts, that are common to all the justifications, with some *dynamic* parts, whose content depends on the output that is returned by the different modules of the framework. As regards the selection of the sentences, we applied some heuristics to improve the internal coherence of the resulting justification. The complete set of sentences was filtered out by removing sentences we labeled as 'non-coherent.' In particular, we maintained just sentences written in the *third-person singular*. In other terms, we prefer sentences like: '*movie has a good plot*' over '*I liked the plot*.' Even if the semantics is almost the same, the first form allows generating more coherent sentences. These heuristics was also applied in similar research (Musto et al. 2020).

Moreover, it should be pointed out that two different strategies are implemented depending on whether the *separate* or *centroid* strategy to rank the excerpts is selected. When *separate* ranking is exploited, one sentence (i.e., that ranked first) is selected for each contextual dimension. Next, all the sentences are put together by exploiting simple connectives, such as adverbs and conjunctions. Conversely, with the *centroid* strategy, the centroid vector of the different contextual situations is first calculated. Then, top-3 sentences are selected.

An example of the resulting justifications for the *toy example* reported in Sect. 3.2 is provided in Table 2. As shown in the table, a different justification—highlighting a different set of characteristics of the item—is provided for each contextual situation. In our opinion, such a diversification can be useful for the target users who are going to assess the goodness of the recommendation they received or to make their decisions.

This intuitive finding will be validated through a user study, which is discussed in the next section.

4 Experimental evaluation

The goal of the experimental session was to answer the following research questions:

- **Research Question 1 (RQ1):** What is the optimal lexicon (i.e., unigrams, bigrams, unigrams+bigrams) that allows identifying suitable review excerpts to be combined in *post hoc context-aware justifications*?
- **Research Question 2 (RQ2):** What is the optimal strategy to combine the review excerpts in a *post hoc context-aware justification* when the contextual situation is composed of more than one contextual condition?
- **Research Question 3 (RQ3):** How effective is our approach, in comparison with both context-aware and non-contextual methods to generate *post hoc justifications*?

In order to answer both the research questions, we arranged a user study based on 273 subjects (male=50%, degree or PhD=26.04%, age≥35=49,48%, already used a RS=85.4%). Users were recruited by exploiting the *availability sampling* strategy.

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

Restaurant	Justification
Company=Partner	You should visit restaurant 'Antiche Mura.' It is very suitable for a dinner with your partner because *music* of the place *is very romantic*. Moreover, the *terrace is very elegant*
Company=Family	You should visit restaurant 'Antiche Mura.' It is very suitable for a dinner with family since *it results as very clean* and location is *very airy*
Company=Friends	You should visit restaurant 'Antiche Mura.' It is perfect for a dinner with your friends since *price is really excellent.* Moreover, *food is very genuine*
Company=Partner Mood = good Meal = dinner (*centroid* strategy)	You should visit restaurant 'Antiche Mura.' It is perfect for the contexts you selected since *music of the place is very romantic.* Moreover, *pizza is very good*
Company=Partner Mood = good Meal = dinner (*single* strategy)	You should visit restaurant 'Antiche Mura.' It is perfect for a dinner with your partner since *music of the place is very romantic.* Moreover, *terrace is very elegant.* Next, people also say that *atmosphere* is very funny and *staff is very helpful*. This makes is suitable given your mood. Finally, based on your meal, you should know that *pizza is very good* and *food is very genuine*

Generally speaking, availability sampling involves selecting a sample from the population because it is accessible, so individuals are selected because they are readily available. This convenience usually translates to easy operation and low sampling costs. Even if other recruiting strategies provide more solid findings, availability sampling is still used in relevant research. In our case, as Alqahtani et al. (2022) recently did, we recruited participants by e-mail and through social networks, in both academic and non-academic contexts. In our case, most of the sample included students, researchers in the area and people not particularly skilled with computer science and recommender systems.

In our study, we evaluated the effectiveness of the strategy in two different domains: *movies* and *restaurant*. As regards the sample of the users, the majority indicated their interest in *movies* as *medium* or *high* (62.78% of the sample). Similarly, 39.57% of the participants stated that they were going out more than two times a week. Metrics we evaluated included *transparency*, *persuasiveness*, *engagement* and *trust*. In particular, *transparency* aims to evaluate whether the justification allows to explain how the system works. *Trust* aims to investigate whether the justifications increase users' trust in the system. *Persuasiveness* evaluates whether the justifications convince users to try or buy the recommended items, and the *engagement* is finally related to the enjoyment concerning the fruition of the item (also defined as *Satisfaction*, in a more general settings). Our selection relied on a subset of the explanation aims defined in previous research (Tintarev and Masthoff 2012). Other aims, such as the *scrutability*, were not considered since they did not fit our settings and the goals of our experimental evaluation.

For all the metrics, the assessment was made through a post-usage questionnaire. The questionnaire is also inspired by similar research carried out in the area of justifications and explanations for recommender systems (Musto et al. 2019, 2020)

4.1 Experimental design

In the following, we describe all the aspects concerning the design of the experiment.
Context definition To start our workflow, for each domain we first identified the *contextual conditions* in which items can be consumed. This work has to be manually carried out, but of course it is possible to exploit previous work in the area where relevant contextual conditions have been defined.

In our case, we selected the context by analyzing related work in the area of context-aware recommender systems in both movies and restaurants domains. In total, three contextual variables were defined for the *movie* domain and 5 contextual variables were defined for the *restaurant* domain. Regarding the movie domain, *mood (great, normal)*, *company (family, friends, partner)* and *level of attention (high, low)* were selected as relevant contextual conditions. Next, as for the restaurant domain, we used again the previously mentioned *mood* and *company*, and we also included *day of the week (weekday, weekend)*, *type of meal (breakfast, lunch and dinner)* and *users' special needs (healthy food, or restrictions)*. Of course, such a design choice can be easily changed or extended in future work.
Data collection Next, we crawled *users' reviews* to feed our components. As regarding the *movie* domain, we carried out a *semi-automatic mapping* based between MovieLens 100k dataset and a set of Amazon reviews.[5] Items were mapped by first matching their names by using some simple scripts. Next, in case of non-exact matching, we used Levenshtein Distance[6] to identify suitable candidates and we manually completed the mapping.

Then, for each contextual condition, a huge set of sentences discussing the item is needed; less popular items (based on IMDB data) and movies with less than 50 reviews were filtered out from the dataset.

As for the *restaurants* domain, we extracted Yelp reviews for the restaurants in the city of Bari (Italy). Again, restaurants having a low number of reviews were filtered out. Of course, in the future, a larger evaluation involving more cities and more restaurants will be carried out. However, in our opinion the current settings provide a sufficient external validity of the findings. Moreover, it is important to emphasize that restaurant reviews were collected for the Italian language. This allowed us to also evaluate the effectiveness of the approach with languages different from English.
Data processing In order to run the NLP steps required by CONTEXT LEARNER and RANKER modules, we exploited tokenization, lemmatization and POS-tagging algorithms available in CoreNLP.[7] Next, the sentiment conveyed by each review was

[5] http://jmcauley.ucsd.edu/data/amazon/links.html—Only the reviews available in the 'Movies and TV' category were downloaded.

[6] https://en.wikipedia.org/wiki/Levenshtein_distance.

[7] https://stanfordnlp.github.io/CoreNLP/.

Table 3 Statistics about the datasets

	Movies	Restaurant
#Items	307	928
#Reviews	153,398	104,000
#Sentences	1,464,593	561,161
#Pos. Sentences	560,817	122,041
Avg. Sent./Item	4770.66	609.01
Avg. Pos. Sent./Item	1826.76	141.22

obtained by using the Stanford Sentiment Analysis algorithm.[8] As we previously explained, only sentences whose sentiment was labeled as *positive* or *very positive* were considered for this step. Some statistics about the datasets is provided in Table 3.

Data annotation Our representations rely on 1,905 annotations we collected for the *movie* domain and 1,710 annotations obtained for the *restaurant* domain, respectively (3,615 in total). Sentences were annotated by three annotators, and a *majority voting* strategy was used to identify suitable contexts. Given that each sentence was evaluated by 3 persons, we collected around 10k annotations in total. Given that 10 persons were involved in the process, each person annotated around 1,000 sentences, on average. This was done in approximately 16 h. The whole process was carried out in two weeks, so the effort per person can be quantified in around one hour per person. No specific tools were used.

Experimental conditions Next, different configurations of the workflow were designed. Such configurations were obtained by varying *vocabulary* of lemmas and *combination* strategy.

As regards the vocabulary of lemmas, we considered three different configurations: one based on the usage of single *unigrams*, one based on *bigrams* and one based on their combination. The difference among the different configurations lies in the selection of lemmas to be encoded in the matrix. In the first case, just single lemmas (e.g., music, terrace, engaging, etc.) are encoded. In the second case, combination of nouns and adjectives (e.g., warm atmosphere) is used. Finally, in the last case we merge them, that is to say, the list of lemmas is based on both unigrams and bigrams. As previously stated, Table 1 contains some of the output of this step and allows a qualitative evaluation of the module.

As for the combination of contextual dimensions, we compared the effectiveness of the previously mentioned *separate* and *centroid* ranking strategy. In total, six different configurations were evaluated.

Experimental protocol To run the experiment, two web applications (one for each domain) were deployed by implementing the methodology described in Sect. 3. Source code of the web applications for the Movie[9] and Restaurant[10] domains has been released.

[8] https://nlp.stanford.edu/sentiment/.

[9] https://github.com/swapUniba/DistributionalSemantics_Movies.

[10] https://github.com/swapUniba/DistributionalSemantics_Restaurants.

Each user interacted with each platform twice. In the first run (from now on, Experiment 1), users were randomly assigned to one of the experimental conditions (based on a specific combination of the parameters). Then, they received a recommendation and evaluated the justifications. Next, in the second run of the experiment (from now on, Experiment 2) the best-performing configuration emerged from the first run was compared to two different baselines in a second experiment.

As for Experiment 1, the following steps were carried out:

1. *Collection of Demographic Data*. After a brief training session, carried out in our laboratory, in which we explained the goals of the experiment and we clarified basic aspects of the methodology, we asked users to give their consent and to provide us demographic information. Finally, they also had to indicate their interest in *movies* and *restaurant* domains.
2. *Selection of the Contextual Situation*. Next, each user indicated the *context* in which the recommended item would have been consumed. This was done by selecting a context among the different contextual conditions we previously indicated (see Fig. 4). Each user could select more than one setting at the same time (e.g., *meal=dinner and company=friends*).
3. *Generation of the Recommendation* Next, based on the contextual situation selected by the user, a suitable recommendation was identified and presented on the user interface. In this case, a context-aware content-based recommendation algorithm fed with review-based features was used. In particular, our approach models the user based on the combination of the vector space representation of the contexts of consumption. The approach works as follows: (a) We provided each item with the concatenation of its reviews. This was the textual content associated with each item; (b) a vector space representation of each item was built, based on these features; (c) based on the contexts of consumption selected by the user, a vector space representation of the user was built as well. In particular, we picked the vector representing the context given as output by our context learner. In case of multiple contexts, the centroid vector is built; (d) the recommendation is returned based on the cosine similarity between the vector representing the user and the vectors representing the items. Of course, this approach does not provide personalized recommendations, since the suggestion is just based on the context of consumption. Accordingly, in order to have more diverse recommendations among the users, we selected the top 5 recommendations obtained by following the previous process and we returned a randomly picked one as a suggestion. However, we point out again that our framework can work with any RS.[11]
4. *Generation of Natural Language Context-aware Justification*. Once the input information is collected, we run our pipeline to generate a justification for the item that also considers its context of consumption. Based on our research questions, a *between-subject* protocol was designed. In other terms, we randomly assigned each user to one of the *six* experimental conditions and we provided her with both the recommendation and the justification. Clearly, both of them are shown

[11] As previously stated, in the future the effectiveness of the methodology on varying of different recommendation algorithms will be assessed.

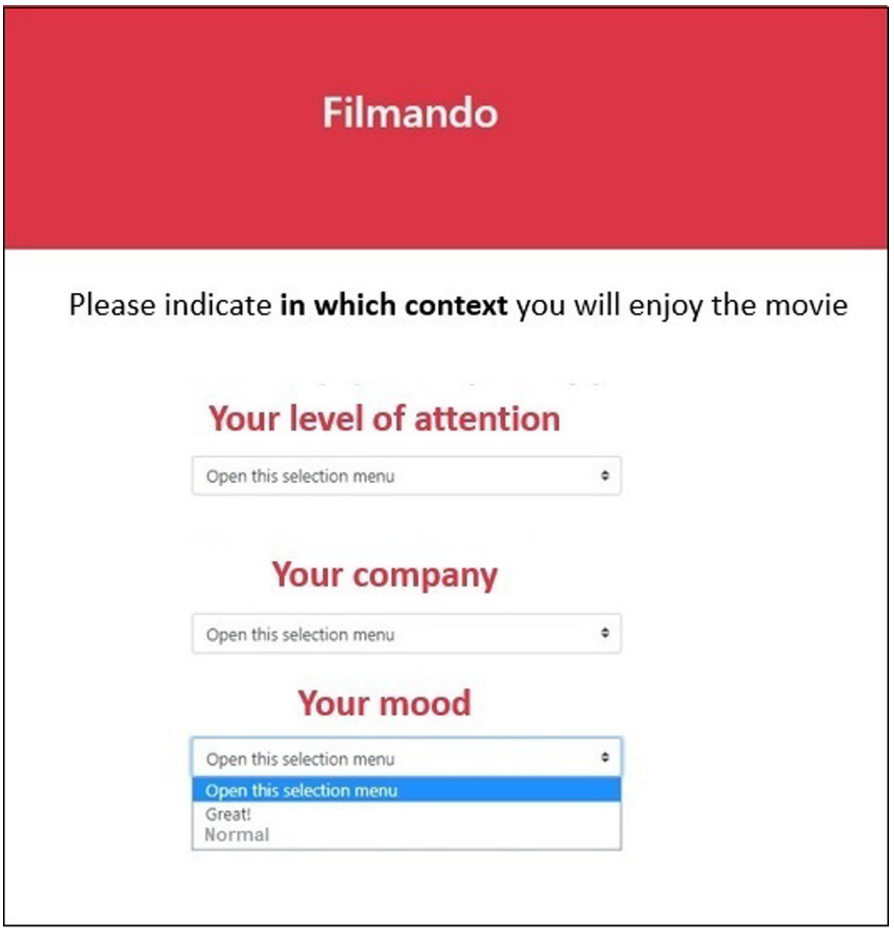

Fig. 4 Context Selection—The screenshot refers to the experiment carried out in the MOVIE domain

on the web application (see Fig. 5), and the users were not aware of the specific configuration they were interacting with (Fig. 6).

5. *Post-usage Questionnaire.* Next, we asked the users to fill in a questionnaire, in order to get some post-usage evaluation. As previously stated, the questions the users had to answer are presented in Fig. 7 and follow those proposed in Tintarev and Masthoff (2012). In particular, users were asked to evaluate the recommendation in terms of *transparency*, *persuasiveness*, *engagement* and *trust*. Each construct was evaluated through a five-point scale (1=strongly disagree, 5=strongly agree). An attention check was introduced to control the quality of the answers.[12] Users who did not pass the attention check were excluded from the analysis.

[12] As Attention Check, we used a common question such as '*Please select strongly agree from the following options*' It was used for all the participants.

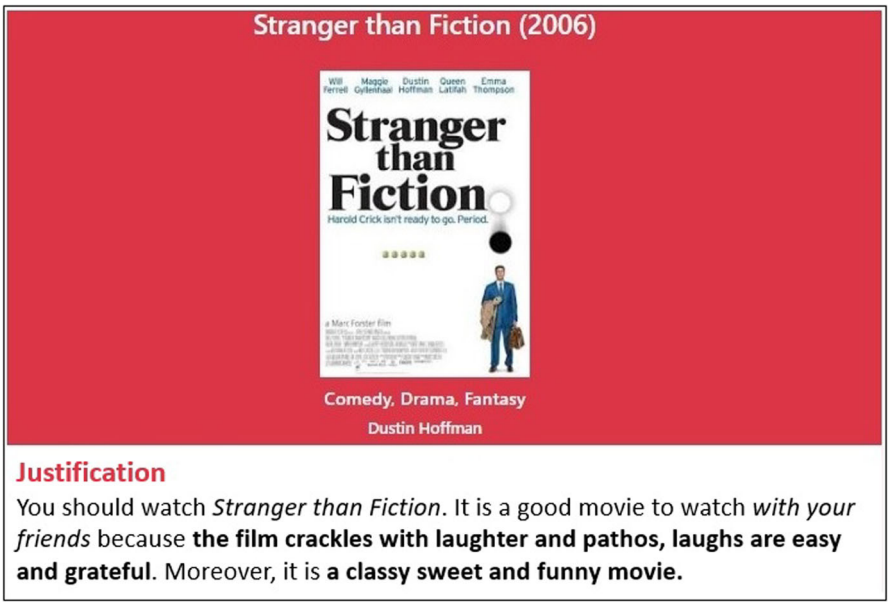

Fig. 5 Recommendation and Justification—The screenshot refers to experiment carried out in the MOVIE domain

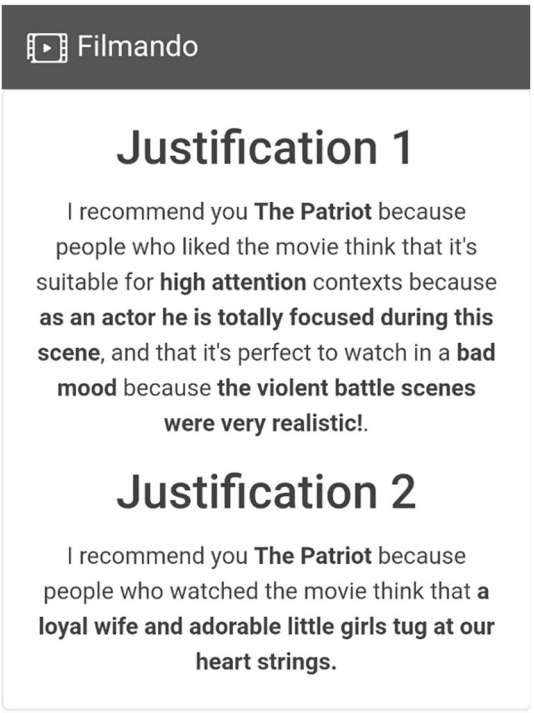

Fig. 6 Comparison to the baseline—The screenshot refers to experiment carried out in the MOVIE domain. In this case, the non-contextual baseline is used as comparison term

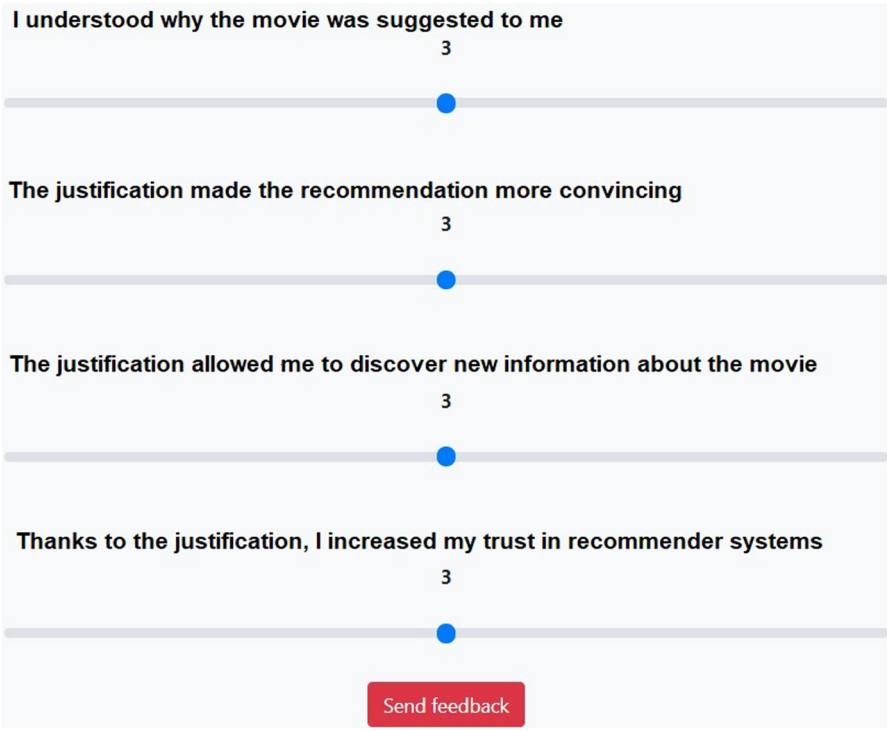

Fig. 7 Post-Usage Questionnaire, Part 1—The screenshot refers to experiment carried out in the MOVIE domain

After Experiment 1, we identified the best-performing configuration by calculating the average score for each of the metrics on varying of the different experimental conditions. Then, after one week, we deployed again the web application by using the best configuration and we run Experiment 2.

As regards Experiment 2, all the steps were repeated. Demographic data were collected again just to check the consistency between the experiment, which was confirmed. The only difference lies in step (3), since the best-performing variant of the framework was compared to two different baselines (which are discussed below), so we run a *within-subject* experiment. In particular, each user was provided with two different justifications (see Fig. 6), i.e., a baseline and one justification generated by using our framework, and the users had to express their preference between the alternatives (see Fig. 8) by using again the post-usage questionnaire.

At the end of the process, the whole protocol was repeated on the second web application. The order in which the domains were presented to the user was randomized. In total, for each configuration of the pipeline we collected between 42 and 49 observations (average = 45.5, standard dev. = 2.87). The number of observation is close to the ideal sample size, indicated in Knijnenburg and Willemsen (2015). As regards RQ1 and RQ2, statistical significance was assessed by carrying out a two-way ANOVA test (with lexicon and combination strategy as *independent variables*), followed by Tukey

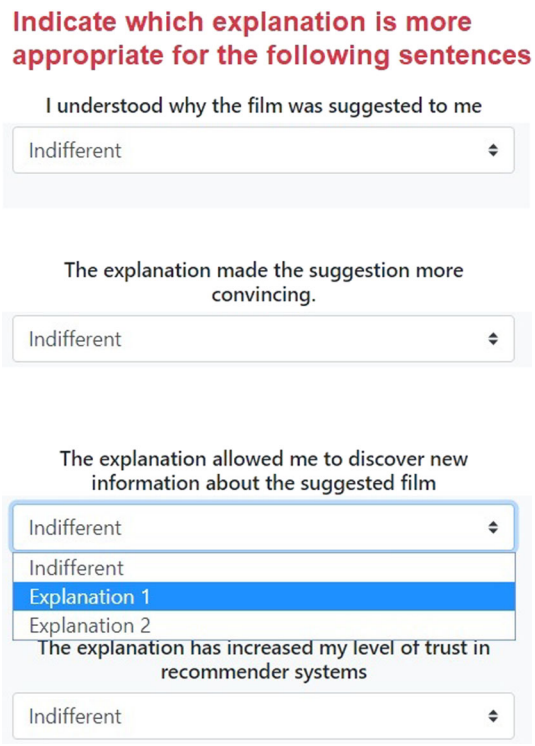

Fig. 8 Post-Usage Questionnaire, Part 2—The screenshot refers to experiment carried out in the MOVIE domain

post hoc tests to compare couples of configurations. As regards RQ3, a Chi-squared test was carried out.

Baselines As previously stated, the selection of the baselines was influenced by the nature and the principles of our methodology. Indeed, we filtered out the available baselines to those that generate a *post hoc* justification. Accordingly, we chose as baselines: (i) a context-aware strategy to generate justifications, which is based on a set of *manually defined relevant terms* for each context. These terms are based on similar research carried out in the area and commonly used heuristics. Given these terms, the justification is obtained through a merge of randomly selected review excerpts containing such terms; (ii) a strategy that generates a review-based justification which is *not context-aware*. Such a strategy first relies on the identification of the aspects of the items that are relevant (i.e., frequently used with a positive sentiment to discuss the items), then combines some reviews excerpts that mention these aspects. The approach follows those presented in Musto et al. (2019, 2020).

Generally speaking, in the first case we aim to investigate to what extent DSMs can be helpful to discover relevant concepts and to automatically learn a vector space representation of the context. Accordingly, we can evaluate whether and how DSMs can be used to learn a lexicon of relevant terms for each context. Next, in the second we investigate to what extent the users appreciate a justification which is *diversified* based on different contexts of usage.

An example of the resulting justifications is provided in Table 4. As shown, in

Table 4 Comparison among our strategy and the baselines we selected

Restaurant	Justification
Company=Partner (our strategy)	You should visit restaurant 'Antiche Mura.' It is very suitable for a dinner with your partner because *music* of the place *is very romantic*. Moreover, the *terrace is very elegant*
Company=Partner (baseline context-aware)	You should visit restaurant 'Antiche Mura.' It is very suitable for a dinner with your partner because *music* of the place *is very romantic*. Moreover, the *atmosphere is very cozy*
Baseline not context-aware	You should visit restaurant 'Antiche Mura.' It is a good choice since *location is very elegant* Moreover, *staff is very helpful.*

Table 5 Results of Experiment 1 for the RESTAURANT domain

RESTAURANT DOMAIN *Metrics/Configuration*	Separate			Centroid		
	Unigrams	Bigrams	Uni+Bigrams	Unigrams	Bigrams	Uni+Bigrams
TRANSPARENCY	3.88	3.85	**4.00**	3.60	3.69	**3.90**
PERSUASION	3.78	3.79	**3.95**	3.52	3.54	**3.71**
ENGAGEMENT	3.76	3.86	**4.03**	3.76	3.65	**3.84**
TRUST	3.66	3.68	**3.90**	3.53	3.49	**3.60**

The best-performing configuration is reported in **bold** and underlined

the table our first baseline uses a fixed lexicon to discover aspects worth to be included. Accordingly, more fine-grained concepts such as 'terrace' (identified by our framework) is replaced by a more common aspect. Next, the second baseline just emphasizes relevant characteristics of the item, without taking into account the contexts of consumption. Results of this comparison are presented next.

4.2 Discussions of the results

Tables 5 and 6 summarize the results of the first experiment. Thanks to the scores that we collected, which represent the average scores provided by the users for each of the previously mentioned questions, we can answer **RQ1**.

By first analyzing the results that we obtained for the *restaurant* domain, two-way ANOVA showed that the combination strategy impacts on all the metrics we considered. In particular, gaps are statistically significant in favor of the *separate* strategy. Accordingly, our results showed that by keeping separate the most relevant sentences we selected for each contextual dimension, the resulting justification makes the recommendation process more transparent ($p < 0.05$, $F = 3.77$), more engaging ($p < 0.03$, $F = 4.50$), more persuasive ($p < 0.01$, $F = 9.08$) and more trustful ($p < 0.05$, $F = 3.92$). As regarding the lexicon, results showed that the best results

Table 6 Results of Experiment 1 for the MOVIE domain

MOVIE DOMAIN *Metrics/Configuration*	Separate			Centroid		
	Unigrams	Bigrams	Uni+Bigrams	Unigrams	Bigrams	Uni+Bigrams
TRANSPARENCY	3.38	**3.81**	3.64	3.35	**3.62**	3.30
PERSUASION	3.56	**3.62**	3.54	3.43	**3.44**	3.26
ENGAGEMENT	3.54	**3.72**	3.70	3.41	**3.51**	3.36
TRUST	3.44	**3.66**	3.61	3.22	**3.48**	3.33

The best-performing configuration is reported in **bold** and underlined

were obtained when both *unigrams* and *bigrams* were used. However, regardless of the numerical gaps, statistical tests did not show a significant difference. Finally, no interaction effect between combination strategy and lexicon emerged. To sum up, the results for this domain showed that the strategy we use to combine the different sentences matter. Conversely, lexicon has a minor impact on the results. This outcome may be probably due to the lower number of sentences available in this dataset. As shown in Table 3, the number of positive sentences for each item is ten times lower w.r.t. the number of sentences we have for the restaurant domain. Accordingly, due to the lower number of available sentences, it is likely that the resulting justifications may not differ among the different lexicons and this may justify the absence of a statistical significance among the configurations.

This result is partially confirmed by the results collected for the *movie* domain. In this case, both the combination strategy and the lexicon have a main effect on transparency and trust of the recommendations. As regards transparency, the gap is statistically significant in favor of *separate* combination strategy ($p < 0.05, F = 3.12$) and with a lexicon based on *bigrams* ($p < 0.01$, $f = 4.27$). Similar outcomes and same p-values ($p < 0.05$) emerge for the trust of the recommendations. Differently from the other dataset, no significant gaps emerged for persuasion and engagement. Even if the results showed that separate sentences and bigrams obtain the overall best results for these metrics as well, gaps are not significant. By running Tukey post hoc tests, statistical significance emerged in the comparison between bigrams and unigrams for transparency ($p < 0.01, T = 2.86$) and between bigrams and unigrams + bigrams ($p < 0.05, T = 2.26$). As regarding trust, gaps were significant when comparing bigrams and unigrams ($p < 0.05, T = 2.31$).

Generally speaking, this experiment provided us with an interesting findings, since results showed that there is a relationship between the amount of available sentences and the quality of the resulting justifications. As for the movie domain, a higher number of sentences-led configuration based on bigrams obtain the best results. Accordingly, it is likely that when single keywords are taken into account, very common and poorly relevant excerpts are included in the justifications. Conversely, by modeling couples of co-occurring relevant lemmas (e.g., 'funny plot,' rather than a simple 'plot') more satisfying justifications that are based on more significant excerpts are generated. Conversely, when the number of sentences is lower, the choice of the lexicon does

Table 7 Results of Experiment 2, comparing our approach *(CA+DSMs)* to a context-aware baseline that does not exploit DSMs *(CA Static)*

Domain	**Restaurants**			**Movies**		
Configuration Metrics	*CA + DSMs* (%)	*CA Static* (%)	*Indiff.* (%)	*CA + DSMs* (%)	*CA Static* (%)	*Indiff.* (%)
TRANSPARENCY	**44.44**	33.33	22.23	**52.38**	38.10	19.52
PERSUASION	**42.86**	33.33	23,81	**54.10**	36.33	19.57
ENGAGEMENT	**41.27**	34.92	23,81	**49.31**	39.23	11.56
TRUST	**41.37**	31.75	26,88	**42.86**	39.31	17.83

The configuration preferred by the higher percentage of users is reported in **bold**

Table 8 Results of Experiment 2, comparing our approach *(CA+DSMs)* to a non-contextual baseline that exploit users' reviews *(review-based)*

Domain	Restaurants			Movies		
Configuration Metrics	*CA + DSMs* (%)	*Review-based* (%)	*Indiff.* (%)	*CA + DSMs* (%)	*Review-based* (%)	*Indiff.* (%)
TRANSPARENCY	**51.49**	28.16	20.35	**53.21**	34.47	12.32
PERSUASION	**54.33**	26.76	18.91	**55.17**	32.33	12.50
ENGAGEMENT	**41.18**	20.88	37.94	**44.51**	32.75	22.74
TRUST	**48.99**	28.75	22.26	**42.90**	42.11	14.99

The configuration preferred by the higher percentage of users is reported in **bold**

not impact the overall quality. Accordingly, we can answer **RQ1** by stating that configurations based on bigrams (or bigrams + unigrams) led to the best results, as long as a sufficient number of sentences annotated with positive sentiment are available. This partially confirms the intuition behind this work, since we thought that the usage of bigrams (which are not exploited for traditional non-context-aware review-based justifications) could have led to better explanations. As regards the choice of the combination strategy, to answer **RQ2** we can state that the choice of keeping each context as separate generally leads to better results. This finding was valid for both the datasets and almost every evaluation metric.

Next, we compared the best-performing configurations emerging from Experiment 1 (that is to say, unigrams and bigrams for the restaurant domain and bigrams for the movie domain, with the *separate* combination strategy) to two different baselines. This comparison allowed us to answer **RQ2**. The results of these experiments are reported in Tables 7 and 8. The first one shows how many users preferred our methodology w.r.t. a context-aware approach exploiting a *fixed lexicon* and w.r.t. a *non-contextual* baseline exploiting users' reviews.

To better explain the goals of this experiment, we can state that the first comparison allows assessing how valid is the intuition of exploiting DSMs to learn a *vector space* representation of contexts, since we compared it w.r.t. a context-aware justification method based on a *fixed* lexicon of relevant terms. Next, through the second comparison

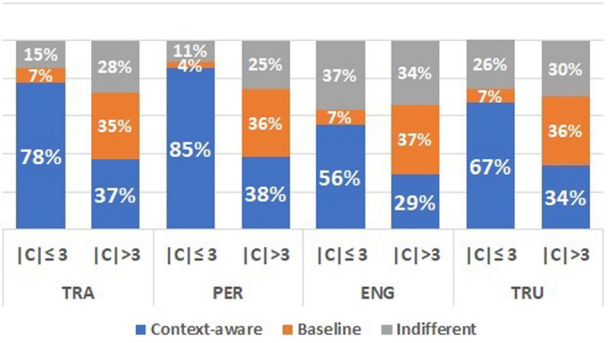

Fig. 9 Distribution of users' preferred justification style on varying of the number of *contexts of consumption* in the RESTAURANT domain. The labels TRA, PER, ENG and TRU refer to Transparency, Persuasion, Engagement and Trust, respectively. For the sake of readability, we did not report fractional parts

we analyze whether the general idea of adapting and diversifying the justifications based on the different contextual situations leads to a better perception of the items.

As shown in tables, both the experiments provided us with encouraging results, since our approach was the preferred one in both the comparisons and for both the domains. By analyzing the results presented in Table 7, gaps are statistically significant for transparency ($p < 0.05$, $X = 3.65$), persuasion ($p < 0.05$, $X = 3.49$) and trust ($p < 0.05$, $X = 3.58$) in the restaurant domain. Similarly, in the movie domain we got statistically significant gaps for transparency, persuasion and engagement ($p < 0.05$).

Moreover, the data we collected showed that the gaps are particularly significant when our methodology is compared to a *non-contextual* baseline. In this case, a statistically significant gap was obtained for all the metrics, with the exception of *trust* in the MOVIE domain.

This finding suggests that the idea of adapting content conveyed in the justifications based on the different *context of consumption* is particularly appreciated by the users.

Finally, in order to deepen the findings presented in Table 8, we further split the results based on the number of *contexts of consumption*, that is to say, the number of contexts the users selected in the context selection phase (see Fig. 5). Of course, the results are always based on the best-performing configuration emerging from Experiment 1.

As shown in Figs. 9 and 10, a large majority of the users preferred context-aware justifications to a non-contextual counterpart when a *lower* number of contexts were selected. In this case, we would have expected that the higher the number of contexts, the larger the number of the users that preferred the context-aware justifications. Contrary to expectations, an opposite behavior emerged. Such a counter-intuitive finding can be explained due to the combination mechanisms that is adopted. When the *separate* strategy is used, a sentence is selected for each context of consumption. Accordingly, it is likely that *very long justifications* are presented to the users when a large number of contexts are selected, and this can lead them to an *information overload* that would make prefer more concise and meaningful justifications, even if

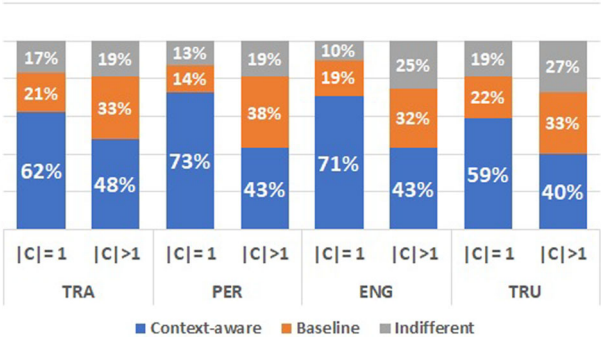

Fig. 10 Distribution of users' preferred justification style on varying of the number of *contexts of consumption* in the MOVIE domain. The labels TRA, PER, ENG and TRU refer to Transparency, Persuasion, Engagement and Trust, respectively. For the sake of readability, we did not report fractional parts

non-contextual. Conversely, with a *limited number of contexts*, the length of the resulting justification is still reasonable and the users showed to prefer such a justification style. Further analyses will be carried out to better investigate such behavior and to propose better strategies to combine different contexts of consumption. Alternatively, it is possible to consider *automatic context selection* techniques: In this context, such methods can be used to identify contextual variables that do not have impact or are not useful, in order to remove unnecessary sentences and make justifications shorter and hopefully more significant.

5 Discussion and limitations

In this section we resume the results by highlighting strengths and weaknesses of the current approach:

- One of the strengths of the approach lies in the flexibility, since it could be also applied to domains different from movies and restaurant. Also domains that are not subjective in nature i.e., computer laptops) could be considered in future experiment. The only requirement that we have is that each recommended item shall be provided with some textual content that describes it.
- Another strength of the approach lies in the independence from the underlying recommendation algorithm. In this case, we have used a simple content-based approach based on the textual content of the reviews. As future work, we will consider to evaluate the approach with different recommendation algorithms since the effectiveness of the justifications may also differ on varying of the algorithm.
- Regardless of the recommendation algorithm, one of the issues of the current method is the fact that justifications are not *personalized*. In other terms, if two users are in the same contextual situation and they receive the same recommendation, they will receive the same justification as well. In the future, the approach could be extended by also considering personal preferences in the pipeline. As an example, aspects that are particularly relevant for the user can be considered and put at the

top of the list. This may be implemented as a weighting factor in the RANKER module.

- It is important to emphasize that some annotation effort in necessary to trigger all the pipeline we have described. In absence of annotations, it is not possible to learn vector space representation of contexts and, in turn, to select relevant sentences in a context-aware fashion. However, as we stated in the previous section, the annotation effort is not high. A limited number of annotations and a limited number of hours are enough to learn accurate representations. As future work, we will also consider whether the annotations collected in one particular domain could be also exploited for different domains as well.
- Our resulting justifications are based on a template. Eventually, template-based justification may result as boring since they always have the same structure. Even if this choice is relatively common and it has already been validated in previous research (Musto et al. 2019), more dynamic generation strategies exist. In future work, we can also consider to adopt more sophisticated generation methods and evaluate them in the same setting.
- Language is a fundamental element of the pipeline. As we explained, our approach can indifferently work with Italian and English, and many more languages following the same principles and same structure could be considered. This is a strength of the framework. However, we did not thoroughly analyzed how language impacts on the resulting justifications, since different outcomes emerged for the different domains as well. This will be done as future work.
- As regarding the experimental evaluation, it is important to point out that we used an availability sampling strategy as recruiting methods. While it can provide accurate and quick results, more reliable sampling mechanisms could be used in the future.
- The post hoc questionnaire is based on just one question per construct. Of course, having more than one question per construct may lead to more reliable results. However, similar research showed that also questionnaires based on a single question per construct can lead to reliable and valid results (Musto et al. 2019). However, as future work, a more extensive evaluation based on multiple questions per constructs could be carried out.
- In the current experiment, we did not ask users about the quality of the resulting recommendations. Even if assessing the quality of the recommendations is not the focus of the work, this could affect the perceptions of the explanations as well. Accordingly, as future work we also plan to introduce some questions related to this aspect, in order to assess whether a relationship between perception of justifications and perception of recommendation exists.
- Finally, further analyses to investigate whether some relationships between the characteristics of the justifications (length, number of sentences, etc.) and the overall perceived quality need to be carried out. This is good direction for future work.

6 Conclusions and future work

In this paper, we introduce a novel strategy to generate natural language review-based justifications that are adapted to the different contextual situations in which an item is consumed. Our approach exploits DSMs to build a lemma-context matrix that encodes the importance of lemmas in each contextual dimension. Such a representation is used to build a vector for each context, which is then used to identify relevant review excerpts that support a recommendation.

The hallmark of this work is the idea of adapting the justifications based on the different contextual situation in which the items will be consumed, which is a new research direction in the area. Moreover, we also designed post hoc justifications that are independent from the underlying recommendation algorithm. As shown in these experiments, users recruited for our studies significantly preferred our justifications w.r.t. non-contextual and less sophisticated baselines in both the domains. As future work, we plan to extend our analysis by also evaluating whether and how other basic properties of the justifications (i.e., size, text complexity, etc.) have influenced the outcome of the study.

Generally speaking, the results confirmed the intuitions behind this work and opens to several future research directions: First, it would be possible to *personalize* our justifications. Personalization can be obtained by also encoding users' interests and preferences in the profile. Moreover, different and more sophisticated strategies to combine different contexts in a single representation could be adopted (e.g., by using other operators to combine the embeddings, such as the *sum*).

We will also consider to compare our approach w.r.t. other baselines, such as a context-aware baseline based on a generic standard context of consumption. Finally, it would be interesting to evaluate the generation of *hybrid* justifications that combine user-generated content (i.e., users' reviews) with descriptive characteristics of the items (i.e., the actor or the director of a movie). This will combine the precision of structured features with the richness of the information obtained by mining users' reviews.

Acknowledgements We acknowledge the support of the PNRR project FAIR—Future AI Research (PE00000013), Spoke 6—Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the Next Generation EU.

Author Contributions Musto contributed to the writing of the entire manuscript, supervised the design of the pipeline for context-aware generation of natural language justifications and was responsible for the experimental setting. Spillo was responsible for the development of the entire pipeline and followed the execution of the experiments. Moreover, he contributed to the writing of some part of the methodological and experimental sections. Semeraro helped with the finalization of the paper, contributed to the writing and supported the initial idea development.

Funding Open access funding provided by Universitá degli Studi di Bari Aldo Moro within the CRUI-CARE Agreement.

References

Adomavicius, G., Bauman, K., Tuzhilin, A., Unger, M.: In: Ricci, F., Rokach, L., Shapira, B. (eds.) Context-Aware Recommender Systems: From Foundations to Recent Developments Context-aware recommender systems, pp. 211–250. Springer, New York, NY (2022). https://doi.org/10.1007/978-1-0716-2197-4_6

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. **23**(1), 103–145 (2005). https://doi.org/10.1145/1055709.1055714

Alqahtani, F., Meier, S., Orji, R.: Personality-based approach for tailoring persuasive mental health applications. User Model. User-Adapt. Interact. **32**(3), 253–295 (2022)

Baral, R., Zhu, X., Iyengar, S., Li, T.: Reel: Review aware explanation of location recommendation. In: Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, pp. 23–32 (2018)

Biran, O., Cotton, C.: Explanation and justification in machine learning: A survey. In: IJCAI-17 Workshop on Explainable AI (XAI), p. 8 (2017)

Chang, S., Harper, F.M., Terveen, L.G.: Crowd-based personalized natural language explanations for recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 175–182 . ACM (2016)

Chen, L., Wang, F.: Explaining Recommendations based on Feature Sentiments in Product Reviews. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces, pp. 17–28. ACM (2017)

Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: Proceedings of the 2018 World Wide Web Conference, pp. 1583–1592 (2018)

Codina, V., Ricci, F., Ceccaroni, L.: Distributional semantic pre-filtering in context-aware recommender systems. User Model. User-Adap. Interact. **26**(1), 1–32 (2016)

Cramer, H., Evers, V., Ramlal, S., Van Someren, M., Rutledge, L., Stash, N., Aroyo, L., Wielinga, B.: The effects of transparency on trust and acceptance of a content-based art recommender. User Model. User-Adapt. Interact. **18**(5), 455–496 (2008)

Deemter, K.V., Theune, M., Krahmer, E.: Real versus template-based natural language generation: a false opposition? Comput. Linguist. **31**(1), 15–24 (2005)

Friedrich, G., Zanker, M.: A taxonomy for generating explanations in recommender systems. AI Mag. **32**(3), 90–98 (2011)

Guidotti, R., Monreale, A., Pedreschi, D.: The ai black box explanation problem. ERCIM News **116**, 12–13 (2019)

Harris, Z.: Mathematical Structure of Language. Wiley, New York (1968)

Haruna, K., Akmar Ismail, M., Suhendroyono, S., Damiasih, D., Pierewan, A.C., Chiroma, H., Herawan, T.: Context-aware recommender system: a review of recent developmental process and future research direction. Appl. Sci. **7**(12), 1211 (2017)

He, X., Chen, T., Kan, M.-Y., Chen, X.: Trirank: Review-aware explainable recommendation by modeling aspects. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1661–1670 (2015)

Hernández-Rubio, M., Cantador, I., Bellogín, A.: A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. User Model. User-Adapt. Interact. **2018**, 1–61 (2018)

Jameson, A., Willemsen, M.C., Felfernig, A.: Individual and group decision making and recommender systems. In: Recommender Systems Handbook, pp. 789–832. Springer (2022)

Knijnenburg, B.P., Willemsen, M.C.: Evaluating recommender systems with user experiments. In: Recommender Systems Handbook, pp. 309–352. Springer (2015)

Lenci, A.: Distributional semantics in linguistic and cognitive research. Ital. J. Linguist. **20**(1), 1–31 (2008)

Li, L., Chen, L., Dong, R.: Caesar: context-aware explanation based on supervised attention for service recommendations. J. Intell. Inf. Syst. **57**(1), 147–170 (2021)
Liu, D., Li, J., Du, B., Chang, J., Gao, R.: Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 344–352 (2019)
Liu, B.: Sentiment analysis and opinion mining. Synthesis lectures on human language technologies **5**(1), 1–167 (2012)
Lowe, W.: Towards a Theory of Semantic Space. In: Proc. of the Twenty-Third Annual Conference of the Cognitive Science Society, pp. 576–581. Lawrence Erlbaum Associates (2001)
Lu, Y., Dong, R., Smyth, B.: Why i like it: multi-task learning for recommendation and explanation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 4–12 (2018)
Manning, C.D.: Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 171–189. Springer (2011)
Manning, C.D., Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn treebank: annotating predicate argument structure. In: Proceedings of the Workshop on Human Language Technology, pp. 114–119. Association for Computational Linguistics (1994)
Mei, L., Ren, P., Chen, Z., Nie, L., Ma, J., Nie, J.-Y.: An attentive interaction network for context-aware recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 157–166 (2018)
Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
Misztal, J., Indurkhya, B.: Explaining contextual recommendations: Interaction design study and prototype implementation. In: IntRS@ RecSys, pp. 13–20 (2015)
Muhammad, K.I., Lawlor, A., Smyth, B.: A live-user study of opinionated explanations for recommender systems. In: Proceedings of the 21st International Conference on Intelligent User Interfaces, pp. 256–260 . ACM (2016)
Musto, C., Lops, P., de Gemmis, M., Semeraro, G.: Justifying recommendations through aspect-based sentiment analysis of users reviews. In: Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, pp. 4–12 (2019)
Musto, C., Narducci, F., Lops, P., De Gemmis, M., Semeraro, G.: 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, pp. 151–154. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2959100.2959173
Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Random indexing and negative user preferences for enhancing content-based Recommender Systems. In: EC-Web 2011. Lecture Notes in Business Inf. Processing, vol. 85, pp. 270–281. Springer (2011)
Musto, C., Narducci, F., Lops, P., de Gemmis, M., Semeraro, G.: Linked open data-based explanations for transparent recommender systems. Int. J. Hum. Comput. Stud. **121**, 93–107 (2019)
Musto, C., de Gemmis, M., Lops, P., Semeraro, G.: Generating post hoc review-based natural language justifications for recommender systems. User Model User-Adapt. Interact. **2020**, 1–45 (2020)
Nakagawa, H., Mori, T.: A simple but powerful automatic term extraction method. In: COLING-02 on COMPUTERM 2002: Second International Workshop on Computational terminology-Volume 14, pp. 1–7 . Association for Computational Linguistics (2002)
Nunes, I., Jannach, D.: A systematic review and taxonomy of explanations in decision support and recommender systems. User Model. User-Adapt. Interact. **27**(3–5), 393–444 (2017)
Reiter, E., Dale, R.: Building applied natural language generation systems. Nat. Lang. Eng. **3**(1), 57–87 (1997)
Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3), 56–58 (1997)
Sato, M., Ahsan, B., Nagatani, K., Sonoda, T., Zhang, Q., Ohkuma, T.: Explaining recommendations using contexts. In: 23rd International Conference on Intelligent User Interfaces, pp. 659–664 (2018)
Schilit, B.N., Adams, N., Want, R.: Context-aware Computing Applications. Xerox Corporation, Palo Alto Research Center (1994)
Shapira, B., Rokach, L., Ricci, F.: Recommender systems handbook (2022)

Sinha, R., Swearingen, K.: The Role of Transparency in Recommender Systems. In: CHI'02 Extended Abstracts on Human Factors in Computing Systems, pp. 830–831. ACM (2002)
Smith, N.A.: Contextual word representations: putting words into computers. Commun. ACM **63**(6), 66–74 (2020)
Staiano, J., Guerini, M.: Depechemood: a lexicon for emotion analysis from crowd-annotated news. arXiv preprint arXiv:1405.1605 (2014)
Tintarev, N., Masthoff, J.: Evaluating the effectiveness of explanations for recommender systems. UMUAI **22**(4–5), 399–439 (2012)
Vig, J., Sen, S., Riedl, J.: Tagsplanations: explaining recommendations using tags. In: Proceedings of the 14th International Conference on Intelligent User Interfaces, pp. 47–56. ACM (2009)
Xia, B., Li, Y., Li, Q., Li, T.: Attention-based recurrent neural network for location recommendation. In: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–6. IEEE (2017)

Giuseppe Spillo is PhD Student at University of Bari, Italy. His research focuses on knowledge-aware recommender systems and in particular, he studies how to combine different and heterogeneous information sources to obtain effective user and item representations. He won the Best Student Paper Award at ACM UMAP 2023 Conference.

Cataldo Musto is Assistant Professor at the Department of Computer Science, University of Bari. His research focuses on the adoption of natural language processing techniques and models for fine-grained semantic content representation in recommender systems and user modeling platforms. He was involved in various national and international research projects that dealt with natural language processing and recommender systems. Since 2009, he published around 70 scientific articles in top venues and journals. He obtained the most inspiring contribution award at UMAP 2013, he got a Best Paper Nominee at RecSys 2016 and a Best Student Paper Award at UMAP 2023. Finally, he regularly acts as a PC member on several top-tier conferences and co-organizes or co-chaires a number of workshops. Recently, he co-organized RecSys workshops about new trends in content-based recsys (2016), UMAP workshops about Holistic User Modeling (2017, 2018 and 2019) and UMAP Workshop on Explainable User Modeling (since 2020). He is one of the authors of the textbook "Semantics in Adaptive and Personalized Systems: Methods, Tools and Applications", edited by Springer.

Giovanni Semeraro is full professor of computer science at University of Bari Aldo Moro, Italy, where he teaches "Intelligent Information Access and Natural Language Processing", and "Programming languages". He leads the Semantic Web Access and Personalization (SWAP) "Antonio Bello" research group. In 2015 he was selected for an IBM Faculty award on Cognitive Computing for the project "Deep Learning to boost Cognitive Question Answering". He was one of the founders of AILC (Italian Association for Computational Linguistics) and on the Board of Directors till 2018. From 2006 to 2011 he was on the Board of Directors of AI*IA (Italian Association for Artificial Intelligence). He has been a visiting scientist with the Department of Information and Computer Science, University of California at Irvine, in 1993. From 1989 to 1991 he was a researcher at Tecnopolis CSATA Novus Ortus, Bari, Italy. His research interests include machine learning; AI and language games; recommender systems; user modelling; intelligent information mining, retrieval, and filtering; semantics and social computing; natural language processing; the semantic web; personalization. He has been the principal investigator of University of Bari in several European, national, and regional projects. He is author of more than 400 publications in international journals, conference and workshop proceedings, as well as of 3 books, including the textbook "Semantics in Adaptive and Personalized Systems: Methods, Tools and Applications" published by Springer. He regularly serves in the PC of the top conferences in his areas and is Program Co-Chair of CLiC-it 2019. Among others, he served as Program Co-chair of CLiC-it 2016, ACM RecSys 2015 and as General Co-chair of UMAP 2013. From 2013, he is the coordinator of the 2nd Cycle Degree Program in Computer Science at University of Bari. He is the coordinator of the 1st edition of the Master in Data Science at University of Bari. He is a member of the Steering Committee of the National Laboratory of Artificial Intelligence

and Intelligent Systems (AIIS) of the National Interuniversity Consortium for Informatics (CINI) and of the Steering Committee of the ACM Conference Series on Recommender Systems.