Natural Language Justifications for Recommender Systems Exploiting Text Summarization and Sentiment Analysis

Cataldo Musto, Gaetano Rossiello, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro

Department of Computer Science
University of Bari Aldo Moro, Italy
name.surname@uniba.it

Abstract. This paper reports and summarizes the methodology presented in [16] and accepted for publication at ACM RecSys 2019[1]. In this work we present a methodology to *justify* recommendations that relies on the information extracted from *users' reviews* discussing the available items. The intuition behind the approach is to conceive the justification as a *summary* of the most relevant and distinguishing aspects of the item, automatically obtained by analyzing its reviews.
To this end, we designed a pipeline of natural language processing techniques including *aspect extraction, sentiment analysis and text summarization* to gather the reviews, process the relevant excerpts, and generate a unique *synthesis* presenting the main characteristics of the item. Such a summary is finally presented to the target user as a *justification* of the received recommendation.
In the experimental evaluation we carried out a user study in the *movie* domain (N=141) and the results showed that our approach is able to make the recommendation process more transparent, engaging and trustful for the users. Moreover, the proposed method also beat another review-based explanation technique, thus confirming the validity of our intuition.

1 Background and Motivations

A research line which is recently emerging in the context of the so-called *Data Explosion* [19] regards the analysis and the exploitation of the information extracted from user-generated content and, in particular, from *users' reviews*. These data are interesting from both a *quantitative* and a *qualitative* point of view, suffice it to say that more than 700 million reviews and opinions are available on TripAdvisor[2]. The positive impact of the information extracted from users' reviews and from user-generated content has been already acknowledged in several scenarios [1,9,14,15]: as an example, *review-aware recommendation models* tend to beat classical recommendation algorithms [3,6,7].

[1] https://recsys.acm.org/recsys19/
[2] https://tripadvisor.mediaroom.com/us

However, differently from most of the current literature, in this work we investigated the impact of the information conveyed by users' reviews in a different scenario, that is to say, the task of *justifying* a suggestion returned by a recommendation algorithm. This topic is particularly relevant due to the recent regulations in the area, as the *General Data Protection Regulation*[3] *(GDPR)*, which emphasized the *users' right to explanation*. Moreover, many studies showed that a higher *transparency* leads to a higher *trust* [20] and to a higher *acceptance* of the recommendations [4].

In a nutshell, our methodology relies on the idea that an effective justification can be conceived as a *summary* of relevant and distinguishing aspects of the item, automatically obtained by analyzing its reviews. To this end, we designed a pipeline of natural language processing techniques to obtain and process reviews excerpts and to generate a unique *synthesis* presenting the main characteristics of the item, which is finally presented to the target user as *justification* of the received recommendation. It is worth to note that we preferred to use the term *justification* over *explanation*, even if they are often used as synonyms. This choice follows the definition provided in [2], where it is stated that a *justification explains why a decision is a good one, without explaining exactly how it was made*, while an explanation is related to the concept of interpretability, that is to say, *if the internal mechanisms as well as the process carried out by an algorithm can be understood by a human.* Accordingly, the strategy we implemented is closer to the idea of justification, since our methodology is more devoted to describe *why* a user would be interested in the item, in order to make a more *informed decision* about consuming the item or not.

For a discussion of the literature in the area we suggest to refer to the original paper [16]. For the sake of shortness, we can state that the distinguishing aspects of the current work lie in:

- The use of text summarization techniques to generate natural language explanations;
- The use of aspect extraction methods to automatically discover relevant and distinguishing aspects of the item, discussed in user reviews.

In the following, we describe the main building blocks of our methodology and we will show the results emerging from the user study in the movie domain. Finally, we will discuss the main outcomes of this work, by also sketching some ideas for future research.

2 Methodology

Our methodology to build *natural language justifications* by exploiting users' reviews relies on four steps: *aspect extraction*, *aspect ranking*, *sentence filtering* and *text summarization*.

[3] http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf

2.1 Aspect Extraction

First, we assume that an effective justification should include relevant and distinguishing traits of the items, which are often discussed in the reviews (with a *positive* sentiment, of course). Accordingly, the goal of the first phase is identifying the aspects that are worth to be included in the justification. Formally, our strategy takes as input a set of reviews $R = \{r_1, r_2 \dots r_n\}$ and produces a set of 4-tuples $\langle r_i, a_{ij}, rel(a_{ij}, r_i), sent(a_{ij}, r_i)\rangle$ representing the review r_i, the *j-th* aspect a_{ij} extracted from the review r_i, the relevance $rel(a_{ij}, r_i)$ of the aspect a_{ij} in the review r_i, and the sentiment associated to that aspect.

To extract *aspects* from reviews, we used the *Kullback-Leibler divergence* [8] (KL-divergence, referred to as δ), a non-symmetric measure of the difference between two distributions. Formally, given two corpora c_a and c_b and a term t, pointwise KL-divergence is calculated as:

$$\delta_t(c_a||c_b) = p(t, c_a) log \frac{p(t, c_a)}{p(t, c_b)} \quad (1)$$

Where $p(t, c_a)$ is the number of occurrences of the term t in corpus c_a. In a nutshell, this measure rely on the idea that the use of language differs when talking about a specific domain with respect to a general topic, thus this method identifies the aspects whose distribution in a specific domain (e.g., *movie reviews*) diverges from that in a general corpus (e.g., the British National Corpus BNC[4]). In other terms, the measure identifies those aspects which are mentioned in the reviews *more often* than usual. Given such a formulation, our strategy for identifying the main aspects mentioned in the reviews follows:

Require: review r_i, general corpus BNC, domain corpus d
Ensure: set of main aspects A and relevance scores
  $A = \{\}$, $T = nouns(r_i) and entities(r_i)$
  for all $t_k \in T$ **do**
    if $\delta_{t_k}(d||BNC) > \epsilon$ **then**
      $a_{ij} \leftarrow t_k$
      $A = A \bigcup \{a_{ij}\}$
      $rel(a_{ij}, r_i) \leftarrow \delta_{t_k}$
    end if
  end for

In our case, given a review r_i, we first extract all the *nouns* by running a POS-tagging algorithm. Next, we calculate KL-divergence for each noun by using as corpora our set of movie reviews and the BNC. The nouns having a KL-divergence greater then a threshold ϵ are labeled as *aspects* and the KL-divergence score is used as relevance score. Finally, we also run a sentiment analysis algorithm to identify the opinion is associated to the aspect a_{ij} in r_i and we stored that value. More details on the algorithms used in this phase are provided in the next Section.

[4] http://www.natcorp.ox.ac.uk/

2.2 Aspect Ranking

Aspect extraction is run over all the reviews contained in R, and a set of aspects is extracted from each review. However, the goal of our strategy is to identify the most relevant aspects that overall describe the item, thus we also designed an *aspect ranking* phase to merge the information we extracted from each single review. Specifically, for each aspect a_j we calculate its *global score* as follows:

$$score(a_j) = \frac{\sum_{i=1}^{N} n_{a_j,r_i} * rel(a_j, r_i) * sent(a_j, r_i)}{N} \quad (2)$$

Our formula gives a higher score to the aspects that are often mentioned in the reviews (n_{a_j,r_i} is the number of the occurrences of a_j in r_i) with a *positive* sentiment. At the end of this step all the aspects are ranked and the *top-K* are labeled as *main aspects*. Typically, the aspects returned by our strategy include concepts such as *actors*, *director*, *story*, *music* and so on.

The *Top-5 Aspects* returned by our algorithm for three different movies are reported in Table 1. Such an output emphasizes how our strategy can make relevant and interesting aspects of the items immediately emerge, since different and distinguishing elements of the movies are highlighted.

Chinatown	**The Ring**	**Titanic**
cast	actor	story
ending	thriller	love
nicholson	effects	effects
performance	horror	picture
story	character	music

Table 1: Top-5 Aspects returned by the Aspect Ranking algorithm for three different movies.

2.3 Sentence Filtering

Once the main aspects are identified, we run a *sentence filtering* phase. The goal of this phase is to filter out *non-compliant* sentences that are supposed to be not useful in the final justification we want to build. In this case we first split each review $r_i \in R$ in sentences $s_{i1} \ldots s_{im}$. Next, we verify the compliancy of each sentence s_i and we maintain only the sentences matching the following criteria: *(i)* s_i contains a *main aspect* $a_1 \ldots a_k$; *(ii)* s_i is longer than 5 tokens; *(iii)* s_i expresses a *positive* sentiment; *(iv)* s_i does not contain first-person personal or possessive pronouns.

The rationale behind these heuristics is straightforward: we want to include in the justification only the sentences including a *relevant aspect* that also express a *positive sentiment* about the item. Moreover, we decided to filter out very short and non-informative sentences as well as those using the *first person*. In this case, the intuition is to prefer excerpts having a more *impersonal* style (e.g., *'The movie has a great cast'*) than those expressing personal opinions (e.g., *I liked the cast of the movie, this is my favourite director*).

2.4 Text Summarization

At the end of the Sentence Filtering a set of potential *candidate sentences* is obtained. Such a set is used to feed a *text summarization* algorithm whose goal is to generate a unique summary to be used as justification of the recommendation. Such a summary is supposed to highlight the main contents discussed in the reviews of the item and to maximize both the *coverage* and *diversity* of the justification. This is done by selecting sentences which cover enough amount of topics discussed in the original reviews and by avoiding the redundancy as well.

To run text summarization we adapted the method described in [18], which proved to be effective in a multi-document summarization task. This makes the approach very suitable for our scenario, since each *review* can be easily considered as a *document*, thus the method can be very effective in summarizing all the reviews excerpts in a single *summary* that highlights the most salient features of the item.

Our approach combines centroid-based text summarization [17], which has the advantage of being unsupervised, with a pre-trained neural language model, such as Word2Vec [10], which is good in transferring information from web-scale textual corpora, and is based on two steps: first, all the information coming from the reviews of an item are condensed in a *centroid vector* which represents a pseudo-review. Next, the main idea is to project both the centroid and each sentence of the reviews in a vector space and to include in the summary only the sentences closer to the centroid.

Formally, given a set of reviews R and its vocabulary V with size $N = |V|$, we first define a matrix $M \in R^{N,k}$, so-called *lookup table*, where the i-th row is a word embedding of size k, $k << N$, of the i-th word in V. The values of the word embeddings matrix M are learned by using Word2Vec. When the lookup table is learned, the summarization method consists of three phases:

Centroid Vector Building. The centroid vector that represents the *meaning* of the reviews is built in two steps. First, the most meaningful words occurring in the reviews (that is to say, those having their $tf * idf$ weight greater than a topic threshold) are selected. Next, the embedding of the centroid is computed as the sum of the embeddings of the top ranked words in the reviews using the lookup table M.

$$C = \sum_{w \in R, tfidf(w) > t} M[idx(w)] \tag{3}$$

In the eq. (3) we denote with C the centroid embedding related to the set of reviews R and with $idx(w)$ a function that returns the index of the word w in the vocabulary.

Sentence Scoring. Given the centroid vector, we need to identify the sentences to be included in the summary. For each sentence in the set of the reviews, we create an embedding representation by summing the vectors for each word in the sentence stored in the lookup table M.

$$S_j = \sum_{w \in S_j} M[idx(w)] \tag{4}$$

In the eq. (4) we denote with S_j the j-th sentence in the set of reviews R. Then, the sentence score is computed as the cosine similarity between the embedding of the sentence S_j and that of the centroid C of the set R.

$$sim(C, S_j) = \frac{C^T \bullet S_j}{||C|| \cdot ||S_j||} \tag{5}$$

Sentence Selection. The sentences are sorted in descending order of their similarity scores. The top ranked sentences are iteratively selected and added to the summary until the limit, in terms of the number of words in the summary, is reached. In order to minimize the redundancy of the information included in the summary, during the iteration we compute the cosine similarity between the next sentence and each one already in the summary. We discard the incoming sentence if the similarity value is greater than a threshold.

At the end of this process a *summary* is produced. An example of the summaries generated for three movies is reported in Table 2. These summaries were used in our experiments as justifications.

3 Experimental Evaluation

The goal of the experimental session was twofold: to evaluate the effectiveness of different configurations of our justifications based on text summarization *(Experiment 1)*, and to compare our methodology to a baseline that exploits users' reviews without text summarization *(Experiment 2)*.

To this end, we designed a user study in the *movie* domain involving 141 subjects. To run the experiment, we deployed a web application[5]. As regards *Experiment 1*, we run the experiment in a *between-subject* fashion, that is to say, each user was randomly assigned to a configuration of our pipeline, and he evaluated the justifications. Clearly, the user was not aware of the specific configuration he was interacting with. Conversely, for *Experiment 2*, we run a *within-subject* experiment, that is to say, all the users were provided with two different explanation styles (*i.e.*, our justifications exploiting text summarization and a review-based baseline presented in [13]).

To run the experiment we built a dataset by mapping MovieLens data with Amazon reviews discussing the movies. The resulting dataset contained 307 movies and 153,566 reviews. The average length of each review was 138.38 words.

[5] http://193.204.187.192:8080/WebLodrecsys_AES

Movie	Justification
Chinatown	*This movie has a decent plot and great acting by Jack Nicholson. Unlike most noir films Chinatown has a great script with fantastic characters, a whirling, looping (but always believable) plot, and one of the best endings of any Hollywood film of any period.*
The Ring	*If you like or love the blood and gore kinds of films, this movie will certainly disappoint you as the focus is on character, story, mood and unique special effects. The Ring is a story about supernatural evil therefore, it is a horror film, done very much in the style of the psychological thriller.*
Titanic	*The highest grossing movie of all time, James Cameron's Titanic follows the love story of Jack and Rose set on the doomed 1912 maiden voyage of the Titanic. This is the greatest love story of our time and a very good drama with great special effects.*

Table 2: Example of justifications generated by our automatic text summarization technique.

In order to evaluate different configurations of our pipeline, we compared six different different alternatives of the algorithm, obtained by varying *number of aspects* and *justification length*. As for the length, we compared *long* and *short* justifications, depending on the overall length of the justification (50 words for short justifications, 100 words for long justifications). As for the number of aspects, we compared the justifications built by including *all* the aspects mentioned in the reviews, the *top-10* aspects and the *top-30 aspects*.

Implementation details. Recommendations were generated by using Personalized PageRank (PR) [5] as recommendation algorithm. As for the Aspect Extraction and the Aspect Ranking, we exploited the algorithms available in CoreNLP[6]. To identify the sentiment conveyed by the single sentences, we used the Stanford Sentiment Analysis algorithm[7]. As regards the Text Summarizer, pre-trained word embeddings learned through Word2Vec were used.

In a nutshell, each user involved in the experiment carried out the following steps:

(1) Preference Elicitation and Generation of the Justifications. We asked users to explicitly rate at least three items, chosen among a randomly generated subset of 20 movies. Next, recommendations were generated.

(2) Between-subject Evaluation through Questionnaires. We asked the users to fill in a questionnaire to evaluate the quality of the justification.

[6] https://stanfordnlp.github.io/CoreNLP/
[7] https://nlp.stanford.edu/sentiment/

Each user was asked to evaluate the previously presented *metrics* through a five-point scale (1=strongly disagree, 5=strongly agree).

(3) Within-subject Evaluation through Questionnaires. We asked each user to evaluate the explanation style they preferred between our methodology and the baseline. As shown in Figure 1, we provided the user with both the explanations in the same screen, and we asked them to select the best one.

Fig. 1: Screenshot of the platform during the within-subject the experiment

Finally, as for the evaluation metrics we used *transparency*, *persuasiveness*, *engagement* and *trust* of the recommendation as the average score collected through the user questionnaires, as in [22].

3.1 Discussion of the Results

The results we obtained for Experiment 1 are reported in Table 3. The values represent the average scores given by the users for that specific metric. The higher the better, with the exception of the effectiveness where the best configuration is the one closer to zero.

Configuration		**Metrics**				
Aspects	*Length*	TRA	PER	ENG	TRU	EFF
All	Short	2.89	2.74	3.26	2.93	0.48
Top-10	Short	2.83	3.06	3.06	2.83	0.89
Top-30	Short	3.16	3.06	2.69	3.19	0.94
All	Long	3.58	3.46	3.29	3.38	0.45
Top-10	Long	**3.95**	**3.64**	**3.37**	**3.55**	0.55
Top-30	Long	3.24	3.18	3.12	3.22	**0.38**

Table 3: Results of Experiment 1. The best-performing configuration for each metric is reported in bold.

The first result emerging from the experiment is that *longer justifications* tend to beat their shorter counterpart. This means that very short justifications cannot convey the information which is needed by the final users to better understand the quality of the suggestions they received. Such a low quality is also confirmed by analyzing the scores we obtained for *short justifications*: they are often below 3.00 (equivalent to *'partially agree'* as answer), which is considered as the minimum score that makes a justification as *acceptable*. Conversely, when longer justifications are produced by our technique, the results we obtained are generally higher. This means that the addition of more sentences can make the justifications and the recommendation process more satisfying, engaging, transparent and trustful for the target users.

As regards the number of *aspects* to be included, the experiment showed that the best results are obtained by exploiting the *top-10 aspects* identified by our aspect ranking module. This is confirmed for all the metrics except of the effectiveness. This means that by selecting a subset of relevant aspects we can provide the user with a summary containing the most relevant and useful information to evaluate the quality of the suggestion. Conversely, when a higher number of aspects is exploited (or even when *all* the aspects are took into account), some noise is probably introduced in the summarization process, thus non-relevant or non-interesting sentences are put in the final justification, and this leads to a decrease in the overall results.

Next, in Experiment 2 we compared our justification based on automatic text summarization to the justifications generated by exploiting a review-based baseline in a *within-subject* experiment. Due to space reasons, we only report the results of the comparison between the best-performing configurations emerging from *Experiment 1* and the baseline. Specifically, we used *long justifications based on top-10 aspects*. In a nutshell, our baseline identifies the most relevant aspects by using our Aspect Extraction and Aspect Ranking strategies and randomly selects a sentence expressing a *positive* sentiment that contains that aspects. For more details on the baseline we suggest to refer to our previous work [13].

Movies	Aspects+Summar.	Aspects	Indifferent
Transparency	**54.55%**	40.91%	4.55%
Persuasion	**77.27%**	13.64%	9.09%
Engagement	**63.63%**	27.27%	9.09%
Trust	**68.18%**	4.55%	27.27%

Table 4: Results of Experiment 2. The configuration preferred by the higher percentage of users is reported in bold.

As shown in Table 4, most of the users indicated that they preferred our methodology based on *automatic text summarization*. It is worth to note that we obtained the higher results for *persuasion* and *engagement*. This is a very encouraging outcome that confirmed the intuition behind this work, since the exploitation of text summarization can help to make interesting information about the recommended item emerge, and this can persuade the user in enjoying the item or can let the user discover new information about the item itself.

4 Conclusions

Overall, we can state the results emerging from these experiments confirmed the effectiveness of the approach, and showed that automatic text summarization can be useful to identify the most relevant aspects of the items and support the suggestions generated by a recommendation algorithm.

As future work we will introduce some strategy to automatically tune the parameters of the model, in order to dynamically select the optimal number of aspects and sentences to be included. Moreover, we will also evaluate more sophisticated techniques for natural language processing, in order to include also entities and bigrams in our justifications, and we will investigate how the effectiveness of the justifications changes on varying of the different algorithms used to generate the recommendation, as those based on Linked Open Data [12], distributional semantics [11] and deep learning techniques [21].

References

1. Balazs, J.A., Velásquez, J.D.: Opinion mining and information fusion: a survey. Information Fusion **27**, 95–110 (2016)
2. Biran, O., Cotton, C.: Explanation and justification in machine learning: A survey. In: IJCAI-17 Workshop on Explainable AI (XAI). p. 8 (2017)
3. Chen, G., Chen, L.: Augmenting service recommender systems by incorporating contextual opinions from user reviews. User Modeling and User-Adapted Interaction **25**(3), 295–329 (2015)
4. Cramer, H., Evers, V., Ramlal, S., Van Someren, M., Rutledge, L., Stash, N., Aroyo, L., Wielinga, B.: The effects of transparency on trust in and acceptance of a content-based art recommender. User Modeling and User-Adapted Interaction **18**(5), 455–496 (2008)
5. Haveliwala, T.H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Trans. Knowl. Data Eng. **15**(4), 784–796 (2003)
6. 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. ACM (2015)
7. HU, Z.k., ZHENG, X.l., WU, Y.f., CHEN, D.r.: Product recommendation algorithm based on users reviews mining. Journal of Zhejiang University (Engineering Science) **8**, 023 (2013)
8. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
9. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining text data, pp. 415–463. Springer (2012)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)

11. 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)
12. Musto, C., Lops, P., Basile, P., de Gemmis, M., Semeraro, G.: Semantics-aware graph-based recommender systems exploiting linked open data. In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. pp. 229–237. ACM (2016)
13. Musto, C., Lops, P., de Gemmis, M., Semeraro, G.: Justifying recommendations through aspect-based sentiment analysis of users reviews. In: Papadopoulos, G.A., Samaras, G., Weibelzahl, S., Jannach, D., Santos, O.C. (eds.) Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2019, Larnaca, Cyprus, June 9-12, 2019. pp. 4–12. ACM (2019). https://doi.org/10.1145/3320435.3320457, https://doi.org/10.1145/3320435.3320457
14. Musto, C., Narducci, F., De Gemmis, M., Lops, P., Semeraro, G.: A tag recommender system exploiting user and community behavior. Recommender Systems & the Social Web (2009)
15. Musto, C., Narducci, F., Lops, P., de Gemmis, M.: Combining collaborative and content-based techniques for tag recommendation. In: International Conference on Electronic Commerce and Web Technologies. pp. 13–23. Springer (2010)
16. Musto, C., Rossiello, G., de Gemmis, M., Lops, P., Semeraro, G.: Combining text summarization and aspect-based sentiment analysis of users' reviews to justify recommendations. In: Bogers, T., Said, A., Brusilovsky, P., Tikk, D. (eds.) Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2017, Copenhagen, Denmark, September 16-20, 2019. pp. 383–387. ACM (2019). https://doi.org/10.1145/3298689.3347024, https://doi.org/10.1145/3298689.3347024
17. Radev, D.R., Jing, H., Sty, M., Tam, D.: Centroid-based summarization of multiple documents. Inf. Process. Manage. **40**(6), 919–938 (2004)
18. Rossiello, G., Basile, P., Semeraro, G.: Centroid-based text summarization through compositionality of word embeddings. In: Giannakopoulos, G., Lloret, E., Conroy, J.M., Steinberger, J., Litvak, M., Rankel, P.A., Favre, B. (eds.) Proceedings of the Workshop on Summarization and Summary Evaluation Across Source Types and Genres, MultiLing@EACL 2017, Valencia, Spain, April 3, 2017. pp. 12–21. Association for Computational Linguistics (2017), https://aclanthology.info/papers/W17-1003/w17-1003
19. Schrage, M.: How the big data explosion has changed decision making. Harvard Business Review (2016)
20. 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)
21. Suglia, A., Greco, C., Musto, C., de Gemmis, M., Lops, P., Semeraro, G.: A deep architecture for content-based recommendations exploiting recurrent neural networks. In: Proceedings of the 25th conference on user modeling, adaptation and personalization. pp. 202–211. ACM (2017)
22. Tintarev, N., Masthoff, J.: Evaluating the effectiveness of explanations for recommender systems. UMUAI **22**(4-5), 399–439 (2012)