

# OAUC at CLEF2016 SBS Lab: Using Appeal Elements to Improve Automatic Book Recommendation - Proof of Concept

Michael Preminger<sup>1</sup> and Gjertrud Fludal<sup>1</sup>

Oslo and Akershus University College of Applied Science

**Abstract.** In this article we describe the OAUC's participation in the CLEF 2016 SBS Search Suggestion track. We are trying to represent appeal elements, used in readers' advisory theory and practice, to see if they can be used in an automatic retrieval and recommendation context. We are still working with the pace appeal element, used in fiction to capture how quickly the buildup of the story or the plot is. New this year is the use of intellectually coded appeal-element data done by EBSCO as part of the NoveList<sup>®</sup> service (our gratitude to EBSCO for providing the data).

## 1 Introduction

As pointed out in [1], There are many qualities to books besides their formal characteristics, such as title, author and subject (the latter being examples of metadata). Books, particularly fiction, also evoke the readers' emotions, which is arguably their major mission. This article continues our exploration of how the emotion-evoking as well as other subtle qualities can be discovered in user generated data and subsequently used in a system for automatic classification of books, as a part of an automatic recommender system. For this year's task we are still focusing on the pace subtle element. The challenge is twofold: try to identify certain emotion waking characteristics of books, and measure whether identification of such characteristics helps us match readers' wishes based on similar characterization of their recommendation requests. We are working on operationalizing the pace using document model creation as well as occurrences of adjectives / adjective types in fast-paced vs. leisurely-paced books.

## 2 Theoretic Approach and related work

Emotion evoking characteristics are properties of books that are not usually a part of the metadata, technically, because they are difficult to trace back. Even though most people might agree on one or the other subtle property of a book, there is potential for dispute

In addition to Saricks work on appeal elements [2], introduced in [1], a couple of works following it up are definately worth mentioning.

## 2.1 Saricks framework of appeal

[2] has developed a framework / terminology that enables librarians, or other reading-promoters, to discuss books through short excerpts, user reviews and the like, boiling down to "appeal". Appeal has a number of elements

**Pace** According to Saricks, *pace* is the most important appeal-element, and has the best potential of distinguishing potential readers. Pace has to do with the build up of the story / plot in a book, and how quickly the reader is drawn into it. Some readers (in some situations) will prefer fast paced books, other will rather endeavour on a slow-paced book. [3] also have a pace value they call "Intensifying", for which we do not have any books in our database.

**Characterization** This element has to do with the introversion or extroversion of the characters in the book. Readers often remember the characters in the book more easily than they remember the plot. Alas, the conception of a well developed character varies greatly among users<sup>1</sup>, making this element less hospitable to analysis of appeal than the case is for the pace element.

**Frame** The frame is about the tone of a book (melancholic, positive), its feeling (funny or romantic), and its atmosphere (menacing or elevating). though difficult to define, this element is often decisive for the reader's choice. The book can be amusing, bleak, bittersweet<sup>2</sup>

**Storyline** The storyline is of course dependent on the previously discussed elements. But typical values<sup>3</sup> will be Issue-oriented, Nonlinear or Open-ended.

## 2.2 Representing and modelling appeal elements

The appeal elements are not directly manifest in the book text, let alone its metadata, and we need to find some representation so that a recommender system can take them into account. To this end we need to find some manifest indicators that can automatically match a recommendation request and a book using the appeal elements as evidence (in addition to other evidence), when recommending a book based on this recommendation request.

Finding and using such indicators is a challenge, which character differs among the elements. Being metadata of different kinds rather than full content, the texts we have are sparse, but on the other hand (for a portion of the books) include *reader reviews*, which should be a condensed summarization of the book done by readers, the target group of a recommender system.

---

<sup>1</sup> [3] lists 30 types of characters that can appear in fiction

<sup>2</sup> [3] lists 58 categories of "Tone".

<sup>3</sup> [3] lists 9 types of storyline

One way of implicitly representing an appeal element (element-name element-value) is, using occurrences of sentences that are characteristic to some value of an appeal element. Feeding these to a Natural Language Processing-system, the NLP system may identify functionally similar sentences in any analyzed book-review to use in the classification of the books. In our implementation, a model of an appeal element is a summary of sentences that are likely to appear in a review of a book that has this value (or valence) of this element. This method has a potential for accuracy, but needs quite a large set of reader reviews given to books with known values (and valences) of the appeal element, and is extremely prone to overfitting. A simpler but less exact method will be identifying single words or word combinations, particularly adjectives used by readers when reviewing books of different values / valences of appeal elements. Such words need somehow to be classified, so that a system looking for appeal elements in reviews has a broader repertoire of words to look for than the one occurring in the training set.

Matching can thereafter be done by attempted applying the same, or a slightly different model to the recommendation request, assuming that a recommendation request and a review belong to the same genre. Here we have several options:

- Retrieving books by a traditional retrieval model (using text-based metadata elements for matching) and then reranking so that books with appeal element values matching that of the request, rank ahead of other books
- Weighing up books with matching appeal at retrieval time
- Traditional retrieval accompanied by pseudo relevance feedback based on the appeal models.

As our current main experimentation line is around pace, we will be more detailed discussing pace, than the other elements.

### 2.3 Pace

Pace can be seen as a binary variable, either "High-paced" ("Fast-paced" in NoveList terminology) or "Low-Paced" ("Leisurely-paced"), making it the easiest element to model and represent, but at the same time less controllable. Saricks poses some questions the answers to which may provide clues as to the pacing:

- Is the book densely written?
- Are there short sentences / short paragraphs, short chapters?
- Is there a straight line plot

## 3 Related work and our approach

Our work belongs in the realm of content based recommender systems, like for example [4]. The main advantage of such systems is their independence of users and their history of reading and recommendations. As such, these system have

High pace	Low pace
Character reveals fast	Character reveals over time
Character reacts to external events	Character only psychologically involved in events
Plot exposed early	Plot exposed over time
Book oriented towards an end	Open ended
Dialog-intensive	Detailed descriptions
Short sentences, paragraphs, chapters	

**Fig. 1.** Characteristics of the pace appeal elements

a better ability to recommend items not yet recommended to anyone, thereby better supporting serendipity. They are also less likely to serve very close material to what a reader already has read, thereby supporting novelty, but are prone to over-specialization, which somehow counteracts the advantages above.

Saricks' framework is reportedly being extensively used in libraries, and in recent years it is starting to gain more systematic use, prominently in a Reader's Advisory resource like NoveList. NoveList<sup>4</sup> is a paid service by EBSCO, marketed towards Reader Advisory (RA) services of libraries, active since the late 90's. Among other book characteristics used for recommendation, They have, since 2010 also been recording and utilizing Saricks appeal elements.

In a more research-related context, [5] has developed a conceptual approach (guiding the current research), of using Saricks elements in book recommendations. As a part of a Phd-work, [6] have experimented with automatic extraction of appeal elements from reviews using rules related to occurrences / co-occurrences of types of words from reader reviews. Both the design approach and the evaluation approach are quite straight forward. The appeal element extraction is a combination of a finite list of words (mostly adjectives) expanded by wordnet-extracted synonyms, and rules for these words' occurrence in the sentences of a review. The rules analyse governor - subordinate relations between pairs of words.

Interestingly, [6] have assessed the quality of their ABET extractor by directly comparing its performance to NoveList's recommendations using appraisals by Amazon Mechanical Turk workers as the gold standard, finding ABET more accurate. They also compared the performance their entire system Rabbit (of which ABET is a component) to other recommendation services by using Mechanical Turk appraisers as gold standard when choosing new books that "best relate" to each one from a sample of ten books. The evaluation strategy taken here is very practical, and the results certainly promising. Still we feel that our challenge here is different, as we wish to match books with recommendation requests (not having other books to relate our recommendations to), and we therefore feel that we need to take a slightly more general approach, which is based on a broader classification of Parts of Speech, particularly adjectives.

<sup>4</sup> <https://www.ebscohost.com/novelist>

Resembling [6], we will also need to take a part / whole approach, trying to see (a) whether our NLP-classification has the potential to elicit individual appeal elements (b) whether it is possible to classify recommendation requests the same way as user reviews (whether or not those two types belong to the same genre) and (c) whether correct identification indeed gives us better recommendations on the basis of textual recommendation requests.

## 4 Data, Experiments and Results

### 4.1 Our Data

The SBS Suggestion Track's (SST) data consist of metadata drawn from LibraryThing and Amazon, describing about 2,8 million books, keyed by their ISBN (meaning the number of distinct works is somewhat lower, as ISBN keys manifestations of works). About half of these, (over 1.3 millions) have reader reviews as a part of their metadata. It is these reviews (free texts) that constitute the most important data of this paper.

In order to prepare the data to adjective based analysis, we have so far been taking the following steps:

- POS-tagging of all free texts of the reviews using the Apache OpenNlp<sup>5</sup>
- Collecting all adjectives, basic (< *JJ* >), comparative (< *JJR* >) and superlative (< *JJS* >)
- normalizing the adjective-forms captured by the POS-tagger, and linking each review to the normalized forms of the adjectives.

As we were preparing this year's experiments, based on the approach we have taken, we have seen that the crunching of the data for preparing the adjective-based is extremely time-consuming and at the moment of writing the data are still in the preparation stage. Therefore we will, in this paper, limit ourselves to experiments based on document categorization.

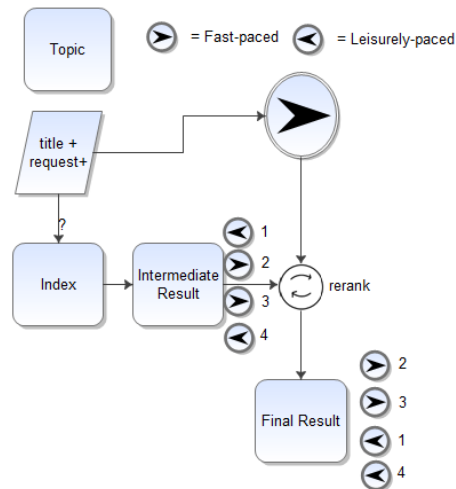
In addition, we have also, with great gratitude, obtained EBSCO NoveList data, categorizing books into value-categories based on several appeal elements.

### 4.2 Our Purpose and Overall Research Design

As an overall, guiding design principle when approaching this issue, we intend to assign values or valences of appeal elements to unseen books (represented by their respective review texts), based on intellectually assigned values to a subset of the books, and building models based on reviews of the latter ones as described in the following subsections. We conduct a traditional text based search into an index created by selected parts of the metadata, and rerank the result so that books with appeal elements *that match the appeal element of the request* are assigned higher prominence in the result set, as depicted in Figure 2.

---

<sup>5</sup> <https://opennlp.apache.org/>



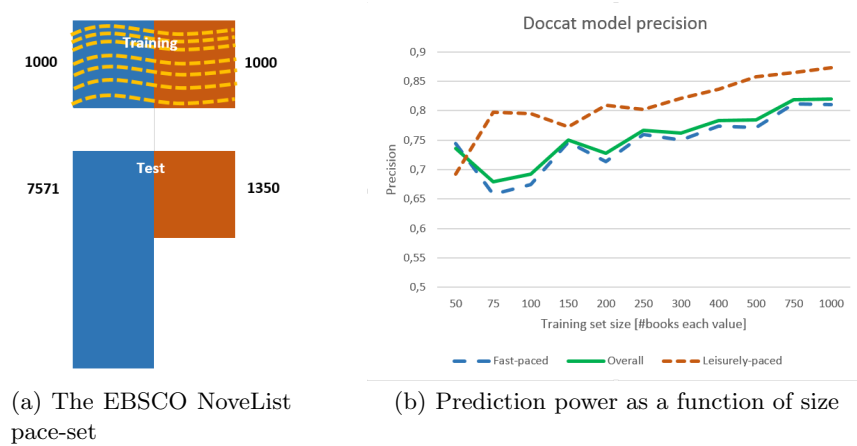
**Fig. 2.** Overall experiment design

**Training the models** The most straight-forward way of training appeal-element categorizers based on existing NLP-tools, is using reviews of books with known values (previously assigned by experts) to build document-categorization models, that can use reviews of unseen books (where available) to classify those into appropriate categories (low vs. high valence, different intervals of element values a.s.o). Classifying the recommendation requests as described in the previous section, can provide us with an additional piece of evidence when matching those requests to books.

In the pace case, with two mutually exclusive values, we are using the Apache OpenNLP (<https://opennlp.apache.org/>) Document Categorizer tool(<https://opennlp.apache.org/documentation/1.6.0/manual/opennlp.html#tools.doccat>) in order to train a number of models for the ability to distinguish (unseen) Fast-paced from Leisurely-paced books by their review texts. The trained models differ by the number of training reviews (respectively 50, 75, 100, 150, 200, 250, 300, 400, 500, 750 and 1000 of each kind, respectively, 2000 reviews used to build the largest model), in order to find out the dependence of the ability to discriminate fast or leisurely paced books on the model size (number of reviews). The 2000 books (1000 of each value) comprising the largest model, constitute a superset of all the other training sets (See illustration in Figure 3(a)). We use the remaining NoveList pace-categorized books as a test set.

In Figure 3(b) we show the prediction power of the model as a function of the number of documents in the training set used to create it. It is interesting to see the trendwise increase in prediction power as a function of the size of the model. This indicates the soundness of the approach, meaning that the pace element tells something about the book, and that the reader review entails a kind of a representation of the pace value. Also interesting is the better prediction

of the leisurely-paced books. This owes to the fact that in this case the test set is closer in size to the training set, so more of the variability of the latter is represented in the former. Paradoxically this also indicates the overall soundness of the approach. We stress that the categorizer is used with default settings, and that work still remains fine-tuning the tool for the categorization task.



**Fig. 3.** The prediction power of the model in terms of prediction precision

**Using Pace in Recommendation / Retrieval** To test the approach’s potential to assist in a recommendation situation, we employ a two-step procedure:

- ranking documents with traditional retrieval methods
- reranking afterwards by matching the paces of the recommendation request and the book reviews

The assumption is that the pace is additional evidence in the ranking process for recommendation satisfaction, and its affect is best measured at the primary part of the ranking list. We reorder the first  $n$  ranked documents returned for each topic, so that documents which pace value matches that of the request (as predicted by the procedure described below), are promoted to the top of the list, keeping their mutual order as in the original list, to that end using a stable sort. For the task of determining the pace of the topic itself, two approaches have been tested:

- Predicting the values of the works attached to the topic in the topic set and applying a majority vote of those as the request value (Figure 4(a))
- Treating the request text as a review text and predicting its value directly. (Figure 4(b))

As the direct prediction strategy (the second above) gives better results in the current experiments, we only report the results of this strategy in the present paper, but hold the other strategy viable for later experiments. The attached work approach (first item above) also has the weakness that not all works attached to the topic have reader reviews associated with them.

**Pace as evidence** Appeal elements, pace among these, are expected to serve as evidence, contributing to the successful recommendation. One way of finding out the optimal contribution would be to rerank upper sublists of various lengths of the original ranked list so that works whose pace match the request pace (predicted by the direct approach from previous section) and measure the performance of these reranked lists up against that of the original list (baseline).

As seen in Figure 5, reranking only the sublists  $n=5-20$  performs better than the baseline (level of evidence 0), whereas reranking longer parts of the list gives unpredictable results, introducing many books with matching pace but otherwise irrelevant to the recommendation request.

## 5 Conclusion

As already explained, we see this as a continuous research endeavor, where the purpose is to directly utilize appeal elements in generating better recommendations based on recommendation requests. We have started out trying to model the pace appeal element in both books' reader reviews and the topics (recommendation requests), trying to see if matching those can give better recommendations. The result so far are promising in that they seem to indicate the overall soundness of the approach, but a better baseline would be needed to test the approach under more realistic conditions.

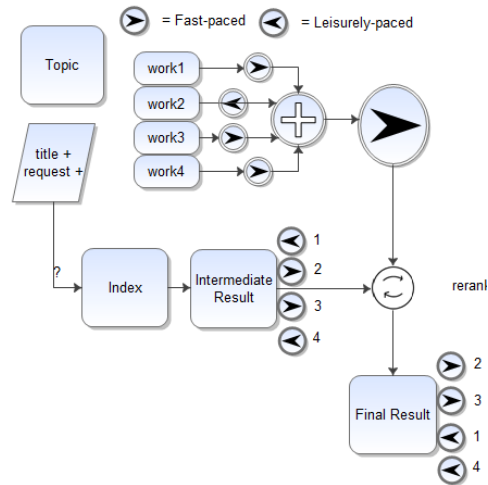
We have so far been using the document categorization strategy as proposed by [5] and [1]. The next endeavour is based on occurrence of adjectives in reviews (indirectly inspired by [6]).

Owing to the relatively large collection of pace values intellectually assigned to books, that we obtained from EBSCO, The current results are remarkably better than the 2015 results. Further use of the data, combining also other appeal elements than pace, indicate that the potential is far from exhausted.

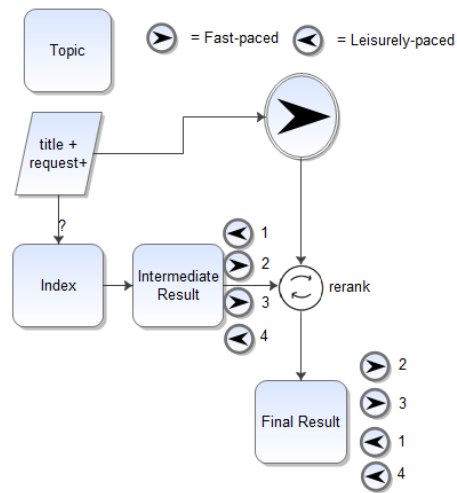
## References

1. Fugleberg, J.R., Preminger, M.: Oauc's participation in the clef2016 sbs search suggestion track. In: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum. Volume 1391., <http://ceur-ws.org> (2015)
2. Saricks, J.: Readers' Advisory Service in the Public Library. ALA editions. American Library Association (2005)



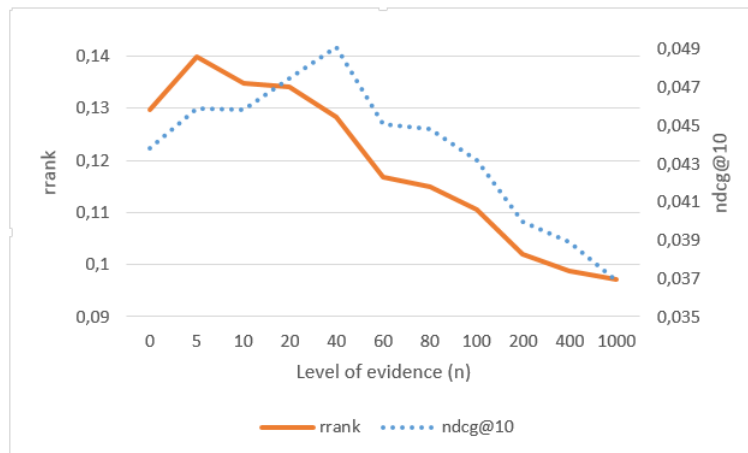


(a) Pace predicted through attached works



(b) Pace directly predicted from request

**Fig. 4.** Overall design - two strategies of predicting request pace



**Fig. 5.** Performance based on level of evidence (length of reordered upper sublist)

3. Caplinger, V., Coleman, E., Coulter, D., Gardner, L., Kage, L., Keyser, C., Morgan, A., Reaser, E., Young, R.: The secret language of books, a guide to appeal. <http://www.ebsco.com/promo/novelist-the-secret-language-of-books> (2015) Promotion Brochure by Ebsco. Accessed: 2015-07-11.
4. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed recommender: Basing recommendations on consumer product reviews. *Intelligent Systems, IEEE* **22**(3) (May 2007) 39–47
5. Fugleberg, J.R.: Automatisk klassifikasjon av bker basert p brukeranmeldelser: Et konsept. Master's thesis, Hgskolen i Oslo og Akershus (2014)
6. Pera, M.S., Ng, Y.K.: Automating readers' advisory to make book recommendations for k-12 readers. In: *Proceedings of the 8th ACM Conference on Recommender Systems. RecSys '14*, New York, NY, USA, ACM (2014) 9–16