A Digital Archive of the *Book of Disquiet* Search and Recommendations

André Filipe Braz dos Santos
andrefbsantos@tecnico.ulisboa.pt
Instituto Superior Técnico

***Abstract*—The Collaborative Digital Archive of the *Book of Disquiet* is platform to show textual fragments of the *Book of Disquiet*, which aims to provide an archive to Fernando Pessoa's work as a representation of the fragments and their interpretations. It also aims to be a platform to create new editions of the the *Book of Disquiet* in a social and collaborative context. In this paper we introduce the development of search and recommendations, based on a set of characteristics, which aims to provide a tool to help users explore the archive and deliver content of the user's interest. The search functionality provides a discovering tool allowing users to explore the archive based on dynamic searches, defined by options selected and configured by them. The recommendation functionality provides a way of discovering the archive according to the user's taste, which means that recommendations are tailored to the users' preferences. The recommendation functionality helps users navigate through similar fragments, based on their taste. Additionally it provides a tool to sort editions based on similarity, creating a semantic edition where fragments' position express their degree of similarity, based on criterias configured by the user. The recommendation follows a content based approach, supported by vector space model, where we translate user's preferences and fragment's properties into vectors and find the degree of interest with cosine similarity. Our implementation of these functionalities follows an approach which not only implements the current search options and recommendations criterias but also allows its extension to support new options and new criterias without changing the overall strategy implemented for search and recommendations.**

***Index Terms*—Search, Recommendations, Fragments, Collaborative Digital Archive, *Book of Disquiet*, Fernando Pessoa**

I. Introduction

The *Book of Disquiet* is an unpublished book composed by a set of fragmentary text written, by Bernardo Soares, one of Fernando Pessoa's heteronyms, from the early 1910 to the year of its death, in 1935.

Until our days, four *Book of Disquiet* were published as a critical vision of the *Book of Disquiet* from their editors, Jacinto Prado Coelho in 1982, Teresa Sobral Cunha in 1990-91, Richard Zenith in 1998 Jerónimo Pizarro em 2010 [1], [2]. Since it was a critical appreciation, they diverge in the amount of fragments, in their order of fragments, following their own thematic and chronological approach, their orthography, Coelho and Pizarro follow Pessoa's orthography while Cunha and Zenith update Pessoa's orthography to their period's orthography. Even the authorship of the fragments is disputed, Cunha presents some of the fragments as been written by Vicente Guedes, another of Pessoa's heteronyms. The *Book of Disquiet* is a book without a consensus about what was Pessoa's vision for this book.

The Collaborative Digital Archive of the *Book of Disquiet* is an web application which aims to work as an archive for the fragments written for Pessoa's book and as a social platform to build new edition of the *Book of Disquiet* [3]. As an archive intends to show the multiple faces of the fragments, from digitalization of the original texts, with its multiple transcriptions, written in the context of the critical editions, codified in TEI [4]. As a social platform, aims to produce new editions of the *Book of Disquiet* built by groups of users, allowing them to express their vision of what the book should be.

The archive was envisioned with the entities shown in figure 1, here we introduce both entities and relations between them [5]. Where we can see how fragments relate to interpretations and sources, heteronym and taxonomies are assigned to fragments and what comprises and edition. Those properties will translate into search options, where users configure an option to find a specific property or a set of specific properties by composing different

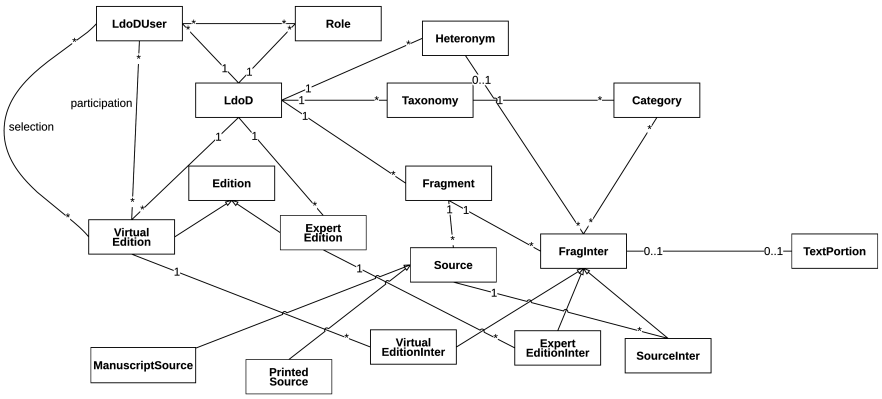

Fig. 1: Class diagram of domain.

options.

The criterias presented to users to configure how suggested fragments are found while they navigate through the archive are also based on the domain model. Those criterias are used to sort an edition, based on the similarity of each fragment to the selected fragment, building a semantic edition where the position of each fragment expresses his degree of similarity to the other fragment. Sorting editions is refined to not only sort editions but also create sections which holds fragments with the same degree of similarity based on a set of recommendation criteria. Sections may also hold sections, expressing a structure of the edition.

Our content based recommendation system supported by vector space model [6], that models user's preferences as vectors. The degree of interest of the user for an item is now expressed by cosine similarity between the user's preference vector and fragment's properties vector.

The implementation of both system follows a set of software engineering good practices in terms of modularity and flexibility, making the implementation open for extension, allowing future implementations of both search options and recommendations' criterias without changing the search or recommendation algorithms.

In section II we define the problem and the goals of our work. In section III we present a solution to address the previously defined problem, which will guide the implementation and integration on the archive related in section IV. In the last section, we reflect upon the work realized and how it can evolved.

II. Problem

The team developing the Collaborative Digital Archive of the *Book of Disquiet* envision a set of functionalities which would help users exploring the archive. Users should be able to explore the archive by searching for fragments, as well as be able to discover new content through recommendations that the user might like.

A. Search

In the field of search, it was envisioned a simple textual search and and advanced search for key properties of the fragments.

Simple search handles the most usual and simple way of searching, therefore it will provide a tool to search for fragments with a set of keywords.

The second level, also known as advanced search was envisioned as a way of combining several different criterias and build a custom search to provide fragments capable of justifying the chosen criteria. Users will arrange and configure several options to find fragments with those properties.

B. Recommendations

On the other hand, recommendations intends to deliver fragments and allow users to discover the archive according to ther taste.

The archive allows users to navigate to other fragments while consulting a fragment. The current navigation method is based on the position of the fragment in the context of editions where it was include, allow users to travel to the previous and next fragment in those editions. We also intend to implement an alternative form of navigation where

the next fragment is the most similar fragment in the context of that edition instead of the fragment following the current fragment in the overall order of the edition. Users should also be able to express their set of criterias which guide the recommendation to find the most similar fragments. This features intends to allow users to explore they archive following a path build specially for them.

It also intends to provide a tool to sort virtual edition by the degree of similarity of fragments to a reference fragment, building semantic edition by a set of chosen criterias. The sort feature should be expanded to not only sort, but also divide a virtual edition in sections, where each section holds fragments with the same degree of similarity to the reference fragment. The last feature will be extended to not only sort, but also cluster fragments with the same value of similarity. All fragments inside a cluster express the same degree of similarity to the reference fragment. It will also be possible to apply clustering measures to a cluster turning a virtual editions in sections with several, simulating a chapter, section, subsection according to a set of criterias chosen by the user to apply to a certain cluster.

III. Solution

A. Simple search

This feature will present a tool to insert a set of keywords which are sent to the server. The servers finds fragments with interpretation with the set of words. The fragments are then presented to the user.

B. Advanced search

Users will build a genetic search through the selection and configuration of multiple options. The allowed options will be heteronym, date, inclusion on critical edition, existence of sources, presence of keywords, inclusion on virtual editions and taxonomies. They will also be allowed to configure the search mode. The conjunctive mode, where a fragment has to satisfy all options. The disjunctive mode, where a fragment only needs to satisfy one option. User selects the options and configure them, the search is sent to the server and returns fragment with interpretations capable of satisfying the options and search mode.

C. Navigation through recommendation

The purpose of navigation through recommendation intends to suggest a fragment similar to the one currently being visualise, therefore, a set of suggestions will be extracted from the user's virtual editions. For each virtual edition which includes the current fragment, the most similar fragment from each virtual edition is found and shown to the user. When the user accepts a recommendation, by travelling to the suggested fragment, he starts a navigation by recommendation, While in this mode, the user will build a recommended edition with the fragments the user has visited while navigating in this mode. While he travels through the recommended edition, the recommended fragment is always a fragment from the virtual edition which the user hasn't seen.

D. Sort editions

When users visualise their virtual editions, they will be able to sort their editions by similarity between fragments of that particular virtual. He will configure a set of recommendation criteria according to their taste and select a fragment to be starting point of the sorted edition. The sorted edition will order fragments according to the similarity to the chosen fragment, following the user's taste, showing the most similarity fragments at the beginning of the edition and the most different at the end. After sorting a virtual edition, they will be able to either save the virtual edition or create a new virtual edition with the new order.

E. Iterative sort editions

The features introduce with sort editions are extended to split an edition in semantic sections of similarity between fragment. The sectioning of editions will be achieved with introduction of sections to the current domain.

F. Recommendation System

The recommendations features presented in the last three points are supported by a content based recommendation system. It will use a memory based strategy supported by vector space model where the similarity is calculated by cosine similarity. This approach translate user's preferences and items' properties into vectors. The similarity is

found through cosine similarity between the user's preference vector and the items' properties vector, funding the likelihood of the user being interested in the item.

1) User's preferences: This recommendation system will use a explicit approach to gather user's preferences. Users will select a fragment and calibrate his degree of interest in the fragments' properties, such as heteronym, date, edition, sources, text and taxonomies. Users' preferences vector will be built from the fragment and will reflect the degree of interest of the user in the fragments properties, according to the calibrate relevance of each property.

2) Properties vectors: The fragment's properties vectors deals with the transformation of fragments properties into vectors. This is achieve through the explicit attribution of a position of the vector to a concrete property. The properties will be built through the concatenation of smaller vectors produced with a proper semantic, where each position express a property inside that semantic. Therefore each one of these small vector will present semantic properties referring to heteronym, date, edition, sources, text and taxonomies.

The properties vector of a fragment's heteronym will express the heteronym associated with their interpretations. The vector has a position for each heteronym in the application and they will contain the value one when there is at least one attribution associated with the heteronym or zero when there isn't a single attribution to that particular heteronym. Figure 2 shows 2 vectors, v_1 is a from a fragment with interpretations only assigned to Bernardo Soares, while v_2 has only interpretations assigned to Vicente Guedes or without assignment.

The date found in the interpretations of a fragment will be expressed in a vector where each position represents a year where the fragment was written. Therefore each position that represents a year found in the fragment's interpretations will appear with the value 1. If we choose to place 0 in the other position, we wouldn't be able to express similarity between fragments written in different years, thus we populate the rest of the vector with values capable of expressing this relation. Starting at a position with a 1, we populate the next and previous positions with 1 minus a predefined decline, the second previous and the second next with 1 minus two times the predefined decline, each position is filled with a value until the cumulative decline reaches 0. From there the rest of the positions are filled with 0. With this strategy we can express similarity between fragments written in different dates. In figure 3 we have a fragment with interpretation written in 1925 and 1917. We can see those positions marked with 1 while the other positions display other values based on the decay.

The inclusion on critical editions will represent the expert interpretations of the fragment. This vector has a position to represent each critical edition in the archive. The position which represents the critical edition where the interpretation belongs, followed by a set o positions to express heteronym and date like described in the previous paragraphs. This edition vector of a fragment will have several copies of this vector and each copy will represent the inclusion in one critical edition. Figure 4 display a vector with edition properties.

The source vector will have three sections to represent each type of source. The first section will have a set o position where the first position appears with 1 when there are manuscript sources or 0 when there isn't. The second position appears with the value 1 when the source has the LdoD mark. The following position express the source information for heteronym and date like previously described. The second section is similar to the previous one, but it marks the existence of typescript source instead of manuscript sources. The last section refers to the printed sources. This section has one position marking the existence of printed sources, which appears with one when there are printed sources or 0 when there isn't. This position is followed by positions to represent heteronym and date like it was previously described. Figure 5 display a vector with source properties.

The theme of the text will be used to express similarity based on the most relevant words of the fragment's interpretations. To calculate the similarity between two fragments, the tf-idf is calculated for each term of the fragment interpretation and the 100 most relevant terms are chosen from each fragment. These terms are the combined into a set. Each position of the vector will represent the term int he same position in the combined set and the value is the tf-idf for that term in the fragment. In figure 6 shows 2 vectors with the correct mapping between term and value, as well was the tf-idf for the term in that fragment.

	Bernardo Soares	Vicente Guedes	Sem Atribuição
v_1	1	0	0
v_2	0	1	1

Fig. 2: Heteronym vector.

	1925	1926	1927	1928	1929
F1	1	0.9	1	0.9	0.8

Fig. 3: Date vector.

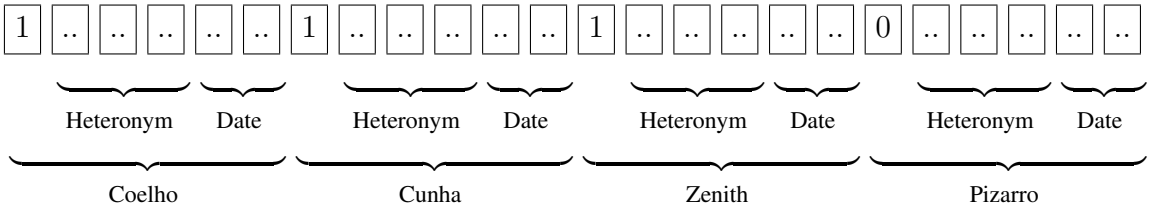

Fig. 4: Edition vector.

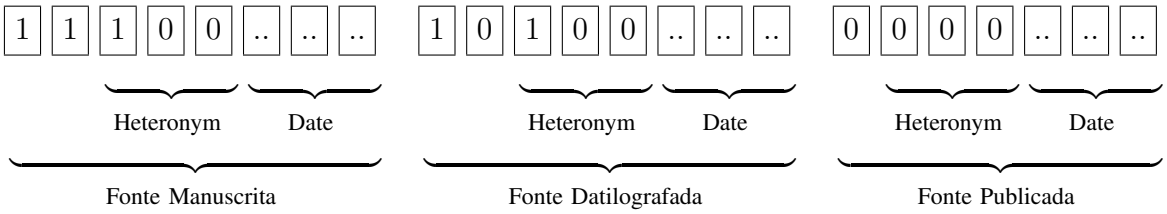

Fig. 5: Source vector.

Taxonomies will be represented in the vector to express the different categories of a taxonomy found in the fragments. The lack of a category will be expressed with a 0 in the vector. When a category is present, we can either chose an binary approach, setting it to 1 or a take advantage of the established domain where each category found in a fragment has a value, which indicates the relevance of the category in the fragment.

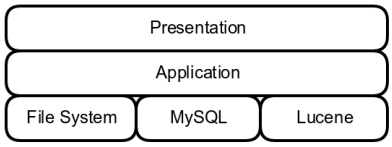

Fig. 7: Overall layer diagram

IV. Integration

The first step to achieve textual search was the inclusion of Lucene to overall scheme of the project. Lucene intends to index the textual content of the fragment's interpretations, when the fragment is uploaded to the application. Therefore Lucene joins the persistence layer, with MySQL to host the domain of the Fenix Framework and the file system to store the facsimiles, like it is displayed in figure 7.

The more detailed view can be seen in figure 8. Here we introduce the the interaction between the different layers of the application. Client-side presentation deals with the interaction with users at the browser level. This layer handles the construction of a search. The dynamic search was achieve with a model view presenter strategy, where users interact with the interface, which the presenter handles and propagates the user's action into the model. The presenter also updates the interface to reflect the current of domain, forming a cycle where the user action is handle the presenter, which manipulates the domain. The domain warns the presenter, which updates the interface to express the domain state. Server-side presentation introduces the parameterization

	Agua	Aguia	Apeadeiro
F1	0	1.5051499783	0
F2	0	0	1.2041199827

Fig. 6: Text vector.

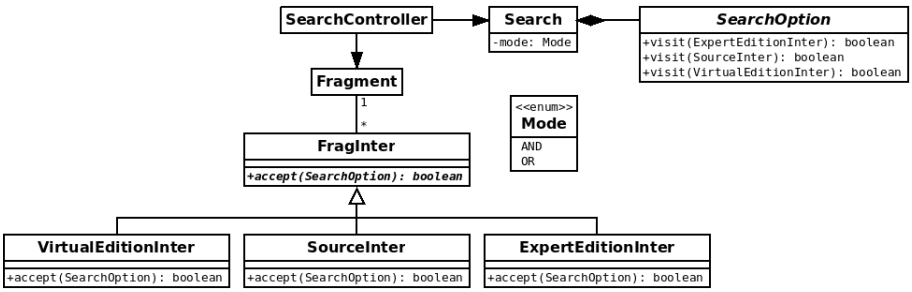

Fig. 9: Class diagram of search.

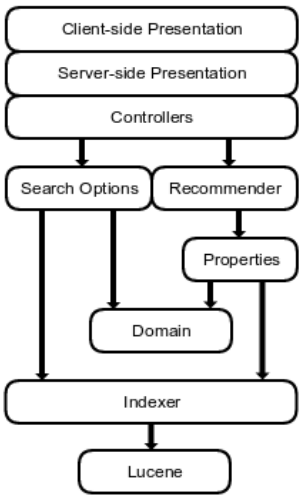

Fig. 8: Detailed layer diagram

of views from templates built in JSP. Controllers handles all the calls from the interface.

The search engine works with several SearchOptions. The classes of this can be seen in figure 9, where the strategy to determine if a fragments respects the search criterias is supported by a visitor which is also displayed. SearchOption offers an interface to build concrete search options.

The classes of the recommendation system can be seen in figure 10. Here we introduce the Recommender hierarchy which intends to offer a common interface for all Recommnders. VSMRecommender is an abstract implementation of a Recommender through VSM and supported by Property to extract properties vectors. Both VSMFRagInterRecommender and VSMFragmentRecommender are concrete recommenders intended to recommend interpretations and fragments. Subclasses of Property were created to implement the transformation of different characteristics into property vectors. VSMFgragInterRecommdner is the one being used to produce recommendations for navigation and establish the similarity sort intended to sort a virtual edition as well as perform their sectioning.

We can also a observe that SearchOptions and Properties interact with the domain the identify characteristics of domain entities, such as interpretaions and fragments, while it also interacts with Indexer to perform textual search. The interface with Lucene is handled through the Indexer.

The domain was also extended to support the weights users define to express their taste, used in the recommendation system. This can be seen in figure 11 with the introduction of RecommendationWeight, to store heteronym, date, edition, sources and text weight, while TaxonomyWeight stores the wights of all the taxonomies in the users virtual editions. In this figure, we also present how a virtual edition is now composed by sections which may contain more sections or virtual interpretations.

V. Related Work

The field of recommendation systems is highly researched and its use is highly desirable in the web, where it aims to provide an information capable of catching the users attention [7], [8], [9]. Online stores use recommendation systems to extrapolate

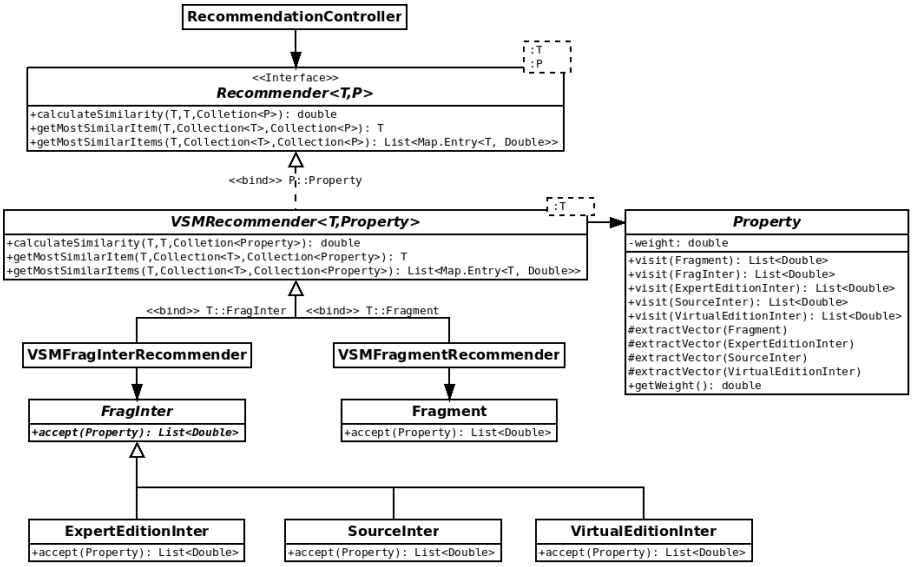

Fig. 10: Class diagram of recommendation.

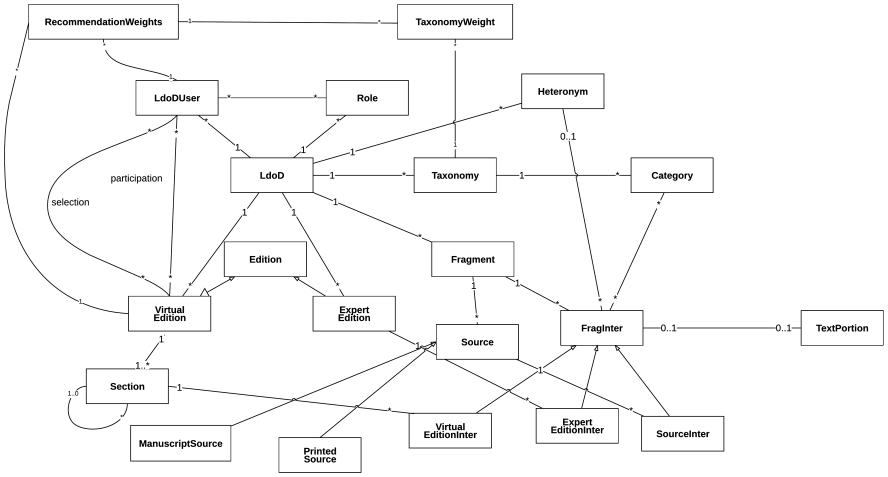

Fig. 11: Class diagram of domain with recommendation weight and sections.

the degree of interest of users in products and recommend items with high chance of being sold.

Recommendation can follow a content based approach, where items are found based on its properties and users interest in these properties [10]. A common approach is the one suggested by vector space model, where vectors are created to express items properties and users' preferences, while computing their similarity between cosine similarity [11].

On the other hand, a collaborative filtering approach [12], intends to discover items based on previous interactions of items and users. To find the user's interest in an item, this approach find the users most similar users, i. e, users which have expressed the same degree of interest in a set of items [13]. From the the group of users, it predicts the user's degree of interest based on the interest express by this group of users.

To produce a prediction, interactions between users and items needs to expressed in ratings [13]. These ratings might be explicit, where users express their interest in items, by classifying them in a scale, e.g. from 1 to 5 or just by placing a like on an item. This approach is highly depended of the users will to provide ratings, but produces more accurate predictions. It can also be implicit, where items are classified based on the nature of the user's interaction, e.g. visualisation of items. While this one doesn't depend on the users willingness

to provide ratings, but their predictions are less accurate.

VI. Conclusion & Future work

The goals of the LdoD team were met through the development of a search and recommendation system, as it was envisioned in the book function of the digital archive. The development of the advanced search system realised the search functions which where required to actively explore the archive. On the other had, the recommendation system finds content according to the user's taste and enriches his interaction with the archive by showing content users will most likely find interesting.

The implementation goal was not only to create a search and recommendation system with several configurable options, but also provide an architecture which could be extended, allowing the implementation of new options. This implementation goal was also met.

The development of new search options and recommendation criterias to deal with more metadata introduced into the digital archive.

The application will grow in its virtual dimension with more virtual editions, therefore a recommendation system for virtual editions might fit into overall lifespan of the archive. If the archive's growth achieves a high volume of users, a collaborative filtering recommendation system will most likely be developed to deal with the recommendation of this growing domain.

References

[1] A. R. Silva and M. Portela, "Tei4ldod: Textual encoding and social editing in web 2.0 environments," *Journal of the Text Encoding Initiative [Online]*, vol. 8, 2015.

[2] M. Portela, "Nenhum problema tem solução: Um arquivo digital do *Livro do Desassossego*," *Estranhar Pessoa com as Materialidades da Literatura*, 2013.

[3] A. R. Silva and M. Portela, "Social edition 4 the book of disquiet: The disquiet of experts with common users," *Journal of the Text Encoding Initiative*, vol. 8, 2014.

[4] "TEI: Text Encoding Initiative," http://www.tei-c.org/index.xml, 2007.

[5] M. Portela and A. R. Silva, "A model for a virtual ldod," *Literary and Linguistic Computing Advance Access*, 2014.

[6] G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[7] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. New York, NY, USA: Cambridge University Press, 2011.

[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2005.99

[9] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends." in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 73–105. [Online]. Available: http://dblp.uni-trier.de/db/reference/rsh/rsh2011.html#LopsGS11

[10] M. Balabanović and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, Mar. 1997. [Online]. Available: http://doi.acm.org/10.1145/245108.245124

[11] R. V. Meteren and M. V. Someren, "Using content-based filtering for recommendation."

[12] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, pp. 81–173, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1561/1100000009

[13] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: Learning new user preferences in recommender systems," in *Proceedings of the 7th International Conference on Intelligent User Interfaces*, ser. IUI '02. New York, NY, USA: ACM, 2002, pp. 127–134. [Online]. Available: http://doi.acm.org/10.1145/502716.502737