SOCIAL TAG PREDICTION: RESOURCE-CENTERED APPROACHES FOR BROAD FOLKSONOMIES

Felipe de Queiroz Badejo Almeida

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Setembro de 2018

SOCIAL TAG PREDICTION: RESOURCE-CENTERED APPROACHES FOR BROAD FOLKSONOMIES

Felipe de Queiroz Badejo Almeida

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Jano Moreira de Souza, Ph.D.

Prof. Daniel Cardoso Moraes de Oliveira, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2018

Almeida, Felipe de Queiroz Badejo

Social Tag Prediction: Resource-centered Approaches for Broad Folksonomies/Felipe de Queiroz Badejo Almeida. – Rio de Janeiro: UFRJ/COPPE, 2018.

XIV, 79 p.: il.; 29, 7cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2018.

Referências Bibliográficas: p. 63 – 77.

1. social tagging systems. 2. multi-label classification. 3. multi-label ranking. 4. tag prediction. 5. tag recommendation. 6. text classification. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

To my family.

Acknowledgments

I would like to thank my family, my advisor Prof. Xexéo, DSc. for inspiring and advising me, Fellipe Duarte, DSc. for patiently answering my questions about scientific research and all staff at COPPE-Sistemas.

In addition, I would not have been able to finish this work without the indirect help of the countless contributors to open-source projects around the world.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PREDIÇÃO DE RÓTULOS SOCIAIS: ABORDAGENS BASEADAS EM RECURSOS PARA FOLKSONOMIAS LARGAS

Felipe de Queiroz Badejo Almeida

Setembro/2018

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Este trabalho aborda o problema de predição de tags (rótulos) em sistemas de tagueamento colaborativo (Social Tagging Systems). É sabido que mecanismos de predição de tags em tais sistemas melhora a usabilidade dos mesmos aumenta a qualidade do vocabulário de tags. Com isso em mente, verificamos a diferença no desempenho de métodos de predição de tags quando aplicados a dois datasets que se diferenciam quanto a número de tags por recurso, quantidade total de tags, quantidade total de recursos, etc. Também analisamos um método específico para predição de tags baseado na quebra de documentos em segmentos. Verificamos se o mesmo generaliza para representações densas de textos. Experimentos são realizados nestes dois conjuntos de dados e os resultados obtidos são relatados.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SOCIAL TAG PREDICTION: RESOURCE-CENTERED APPROACHES FOR BROAD FOLKSONOMIES

Felipe de Queiroz Badejo Almeida

September/2018

Advisor: Geraldo Bonorino Xexéo

Department: Systems Engineering and Computer Science

This work addresses the problem of how to predict tags that will be assigned by users in Social Tagging Systems. It is widely known that tag prediction functionality helps promote system usability and increase the quality of the tag vocabulary in use. With that in mind, we verify the difference in performance of several label ranking techniques on two datasets, which differ from each other in several key metrics such as the average number of tags per resource, tag vocabulary length, total number of resources, etc. We also analyze a specific label ranking technique, namely MIMLSVM. We verify whether it generalizes to dense text representations in addition to traditional sparse ones. Experiments are conducted on the two datasets and results are analyzed.

Contents

List of Figures **xi**

List of Tables **xiv**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Social Tagging Systems . . . 2
1.3 Problem scope . . . 2
1.3.1 Binary Tag Assignment Model . . . 4
1.4 Methodology . . . 5
1.4.1 Literature Review . . . 5
1.5 Document structure . . . 6

2 Social Tagging **7**
2.1 Examples of Social Tagging Systems . . . 7
2.1.1 MovieLens . . . 7
2.1.2 StackOverflow . . . 8
2.2 Social Tagging and Folksonomies . . . 10
2.3 Narrow and Broad Folksonomies . . . 12
2.4 Other Aspects . . . 12
2.4.1 Tag Stabilization and Convergence . . . 12
2.4.2 Effect of tag suggestion on STSs . . . 13

3 Related Work **14**
3.1 Introduction . . . 14
3.2 Resource-centered Methods . . . 15
3.2.1 Association Rule Mining . . . 15
3.2.2 Content-based tag propagation . . . 16
3.2.3 Resource-based tag propagation . . . 17
3.2.4 Multi-label Classification/Ranking . . . 18
3.2.5 Methods based on Topic Modelling/Tensor Factorization . . . 21
3.2.6 Graph-based . . . 22

3.2.7 Other . . . 23
3.3 Other Aspects . . . 25
3.3.1 Data Representation . . . 25
3.3.2 Clustering . . . 26

4 Proposal and Experiment Outline **27**
4.1 Proposal . . . 27
4.2 Datasets . . . 29
4.2.1 Dataset 1: Delicious t-140 . . . 29
4.2.1.1 Construction . . . 29
4.2.1.2 Preprocessing . . . 30
4.2.2 Dataset 2: Movielens 20M + IMDB Synopses . . . 31
4.2.2.1 Construction and Preprocessing . . . 33
4.3 Experiment Outline . . . 34
4.3.1 Project Structure . . . 34
4.3.1.1 Frameworks and Libraries used . . . 34
4.3.2 Method Selection . . . 35
4.3.3 Hyperparameter Tuning . . . 35
4.3.4 Metrics and Evaluation . . . 36
4.3.4.1 Average Precision and Mean Average Precision . . . 36
4.3.4.2 Micro-Averaged F1 @k . . . 37

5 Experiments **39**
5.1 Experiments for Proposal 1 . . . 39
5.1.1 TF-IDF weighted Bag-of-words Features, Binary Relevance + Linear SVM Classifier . . . 39
5.1.1.1 Results on Dataset 1 . . . 40
5.1.1.2 Results on Dataset 2 . . . 40
5.1.1.3 Discussion . . . 40
5.1.2 TF-IDF weighted Bag-of-words Features, k-Nearest Neighbours Classifier . . . 41
5.1.2.1 Results on dataset 1 . . . 42
5.1.2.2 Results on Dataset 2 . . . 42
5.1.2.3 Discussion . . . 42
5.1.3 TF-IDF weighted Bag-of-words Features, Topic Distances . . . 43
5.1.3.1 Results on dataset 1 . . . 43
5.1.3.2 Discussion . . . 44
5.1.4 TF-IDF weighted Bag-of-words Features, Topic Words . . . 44
5.1.4.1 Results on dataset 1 . . . 45
5.1.4.2 Results on Dataset 2 . . . 45

5.1.4.3 Discussion . . . 46
5.1.5 LDA Topic Probabilities, k-nearest Neighbours Classifier . . . 46
5.1.5.1 Results on dataset 1 . . . 47
5.1.5.2 Results on Dataset 2 . . . 47
5.1.5.3 Discussion . . . 47
5.1.6 LDA Topic Probabilities, SVM classifier . . . 48
5.1.6.1 Results on dataset 1 . . . 48
5.1.6.2 Results on Dataset 2 . . . 49
5.1.6.3 Discussion . . . 49
5.1.7 Final Results and discussion . . . 50
5.2 Experiments for Proposal 2 . . . 51
5.2.1 MIMLSVM with IDF weighted Bag-of-words features . . . 53
5.2.1.1 Results on Dataset 1 . . . 54
5.2.1.2 Results on Dataset 2 . . . 54
5.2.1.3 Discussion . . . 54
5.2.2 MIMLSVM with LDA Topic Probabilities as Features . . . 55
5.2.2.1 Results on Dataset 1 . . . 55
5.2.2.2 Results on Dataset 2 . . . 56
5.2.2.3 Discussion . . . 56
5.2.3 MIMLSVM with IDF-weighted Bag-of-embeddings Features . 57
5.2.3.1 Results on dataset 1 . . . 57
5.2.3.2 Results on Dataset 2 . . . 57
5.2.3.3 Discussion . . . 58
5.2.4 Final Results and discussion . . . 58

6 Conclusion and Future Work **59**
6.1 Conclusion . . . 59
6.1.1 Proposal 1 . . . 59
6.1.2 Proposal 2 . . . 60
6.2 Threats to Internal and External Validity . . . 60
6.3 Future Work . . . 61
6.3.1 Alternative similarity metrics for clustering multi-instances . . 61
6.3.2 Alternative clustering algorithms . . . 61
6.3.3 Other classifiers for MIMLSVM . . . 61
6.3.4 Adapting algorithms from Computer Vision to Natural Language Processing . . . 61

Bibliography **63**

A Code Layout **78**

List of Figures

2.1 The MovieLens website supports tagging; any user can add their own tags and view tags assigned by other users to a particular resource. Retrieved from https://movielens.org/movies/54286 in January 2018. 8

2.2 The StackOverflow website also supports tagging, but only a single set of tags is shown, namely the tags assigned by the resource's original owner (and maybe edited afterwards). Retrieved from https://stackoverflow.com/questions/231767/what-does-the-yield-keyword-do in January 2018. 9

2.3 Tags are also used to help drive Stackoverflow's incentive mechanisms; tag medals are awarded for activity related to a certain tag. Retrieved in January 2018. (Blur is used to protect the user's privacy) 10

2.4 A folksonomy can be represented as a tripartite hypergraph, where three-way hyperedges connect users, tags and items. Adapted from RAWASHDEH *et al.* (2013). (Best viewed in colour). 11

2.5 Retrieved from https://www.flickr.com/search/?tags=afghanistan in January 2018. 11

2.6 Tag distributions for two resources on Delicious.com. Tag proportions reach equilibrium after around 100 tag assignments. Adapted from GOLDER & HUBERMAN (2005) 13

3.1 When each label prediction is given a score, we can choose a threshold k and return only the top k labels, as ranked by score. 18

4.1 Distribution of the number of unique tags assigned to each document in the Delicious t-140 dataset (after pruning and preprocessing). . . . 31

4.2 Distribution of the number of documents each tag was assigned to in the Delicious t-140 dataset (after pruning and preprocessing, not counting multiple assignments). 31

4.3 Distribution of the number of unique tags assigned to each document in the Movielens 20M + IMDB Synopses dataset (after pruning and preprocessing). 33

4.4 Distribution of the number of documents each tag was assigned to in the Movielens 20M + IMDB Synopses dataset (after pruning and preprocessing, not counting multiple assignments). 33
4.5 Comparing choice of hyperparameters for feature extraction. Using Dataset 2 for illustrative purposes. 36

5.1 Results of applying Binary Relevance + Linear SVM with TF-IDF features on the Delicious t-140 Dataset (validation set scores shown) . 40
5.2 Results of applying Binary Relevance + Linear SVM with TF-IDF features on the Movielens Dataset (validation set scores shown) . . . 40
5.3 Binary Relevance, Linear SVM with TF-IDF features: Compared results (validation set scores) 40
5.4 Applying k-NN on the Delicious Dataset, using TF-IDF weighted bag-of-words representation (validation set scores shown) 42
5.5 Applying k-NN on the Movielens Dataset, using TF-IDF weighted bag-of-words representation (validation set scores shown) 42
5.6 k-NN with TF-IDF features: Compared results (validation set scores) 42
5.7 Applying Topic Distances on the Delicious Dataset, with varying values for the choice of LDA components 43
5.8 Applying Topic Distances on the Movielens Dataset, with varying values for the choice of LDA components 44
5.9 Topic Distances: Compared results (validation set scores). Best and worst results for each Dataset shown for comparison. 44
5.10 Applying Topic Words on the Delicious Dataset, with varying values for the choice of LDA components (validation set scores shown) . . . 45
5.11 Applying Topic Words on the Movielens Dataset, with varying values for the choice of LDA components (validation set scores shown) . . . 45
5.12 Topic Words: Compared results (validation set scores) using the best choice for the number of components. 46
5.13 k-Nearest Neighbor classifier on the Delicious dataset, using LDA topic probabilities as features (validation set scores shown). 47
5.14 k-Nearest Neighbor classifier on the Movielens dataset, using LDA topic probabilities as features (validation set scores shown). 47
5.15 k-NN using LDA features: Compared results (validation set scores). . 47
5.16 SVM classifier on the Delicious dataset, using LDA topic probabilities as features (validation set scores shown). 48
5.17 SVM classifier on the Movielens dataset, using LDA topic probabilities as features (validation set scores shown). 49
5.18 SVM using LDA features: Compared results (validation set scores). . 49

5.19 Full comparison of all techniques used for proposal 1 (validation set scores). 50
5.20 Multi-instance learning works by representing a single example as multiple instances. 51
5.21 Original algorithm, devised by ZHANG & ZHOU (2006), transforms an MIML problem into either a SIML or an MISL problem, using MIMLSVM and MIMLBOOST techniques, respectively. 52
5.22 MIMLSVM classifier applied on the Delicious dataset, using TF-IDF weighted bag-of-words features (validation set scores shown). 54
5.23 MIMLSVM classifier applied on the Movielens dataset, using TF-IDF weighted bag-of-words features (validation set scores shown). 54
5.24 MIMLSVM with TF-IDF features: Compared results (validation set scores) 54
5.25 MIMLSVM classifier applied on the Delicious dataset, using LDA topic probabilities as features. (validation set scores shown) 55
5.26 MIMLSVM classifier applied on the Movielens dataset, using LDA topic probabilities as features features. 56
5.27 MIMLSVM with LDA features: Compared results (validation set scores) 56
5.28 MIMLSVM classifier applied on the Delicious dataset, using IDF weighted bag-of-embeddings features (validation set scores shown). . . 57
5.29 MIMLSVM classifier applied on the Movielens dataset, using IDF weighted bag-of-embeddings features (validation set scores shown). . . 57
5.30 MIMLSVM with IDF weighted bag-of-embedding features: Compared results (validation set scores) 58
5.31 Full comparison of all techniques used for proposal 2 (validation set scores). 58

List of Tables

1.1 Broad and narrow folksonomies 3
1.2 Tag prediction approaches, classified according to the information they use 3

3.1 Approaches to tag prediction, classified by techniques used 25

4.1 Dataset Statistics: Delicious t-140 (after pruning and preprocessing) . 30
4.2 Dataset Statistics: MovieLens 20M + IMDB Synopses (after pruning and preprocessing) 32

5.1 Compared dataset statistics (after pruning and preprocessing) 50

Chapter 1

Introduction

In this chapter we will introduce the subject matter of this dissertation, namely Tag Prediction under Social Tagging Systems (STSs) and provide some insight into the problem scope and our line of research.

1.1 Motivation

The motivation for this work is twofold.

Firstly, it is easy to see that naïve methods of organization may be hard to use in complex, real-world systems. Tags are one way to help users and administrators better organize and reason about concepts and/or resources in many such systems.

Take the following example: You organize scientific articles into folders and you have a new article called *The History of Football in Europe*. Should you place it under *"history"*, *"sports"* or *"Europe"*? Maybe place a copy under each folder? Create a new folder called *"Europe_Sports"* instead?[1] Clearly, assigning multiple labels or *tags* to each resource is an elegant way of solving this problem.

Secondly, it is well established that suggesting tags to users promotes faster convergence to a common vocabulary (DATTOLO *et al.*, 2010; HASSAN *et al.*, 2009; MARLOW *et al.*, 2006) and increases the likelihood that resources are tagged (DATTOLO *et al.*, 2010; FLOECK *et al.*, 2010). Additionally, it has been claimed that identifying good tags from a set of recommended tags is orders of magnitude less demanding than coming up with good tags without intervention (MARINHO *et al.*, 2012).

Since there are massive amounts of data available from Social Tagging Systems (STSs), it's natural to think of a data-driven, machine-learning based method to pursue that goal. In addition to being a worthy research problem from a theoretical

[1] Note that similar classification schemes have already been in use for some time in places such as libraries. Some noteworthy examples include the Dewey Decimal System and the Library of Congress Classification methodologies, both in use for more than a century.

standpoint, good tag prediction techniques could also effectively help people in the real world navigate these online communities.

1.2 Social Tagging Systems

The massive expansion experienced by the Internet and web communities in the last decades has undoubtedly had a large effect on how we live our lives. Nowadays, we have access to many kinds of services on the web, such as search engines, email messaging, online purchases, and so on.

However, some of the most widely used websites are places where people interact with digital resources and with other human beings. Among these we could cite websites such as Facebook, Twitter, Youtube, StackOverflow, Quora, LinkedIn, Reddit, Sina Weibo and many others.

These online communities form what is commonly referred to *social media* or *social networking services*, or SNSs. (AMICHAI-HAMBURGER & HAYAT, 2017; OBAR & WILDMAN, 2015)

Moreover, some of these so-called social media services support *tags*, which are user-given, generally free-form, keywords used to help categorize resources MATHES (2004). These systems are called **Social Tagging Systems** or **STS**s.

1.3 Problem scope

When considering Social Tagging Systems, one can envision many different problems and areas where scientific knowledge and research could be put to use. For this reason, we chose to address the problem of how to correctly predict which tags will be used to describe a given textual object in such a system.

Since this problem touches upon many areas of scientific knowledge, such as machine learning, natural language processing (for textual resources), computer vision (similarly, for images and visual objects), recommender systems and so on, it is necessary to further limit our scope in a more precise manner.

Firstly, one may create a distinction between **broad and narrow folksonomies**.[2] **Broad folksonomies** are those where not just a resource's owner but the whole community of users may assign tags to any one resource available on the system. **Narrow folksonomies**, on the other hand, only allow items to be tagged once, by the person who has first added that particular item to the system (i.e. that item's owner). A summary can be seen on the following table:

[2]Following commonly-cited sources such as WAL (2005a)

Table 1.1: Broad and narrow folksonomies

Broad folksonomies	Anyone can assign tags to any resource, and all tags are viewable by everyone using the system.
Narrow folksonomies	Only the owner of a given resource may assign tags to it. Other users can view them but cannot add their own.

Secondly, many authors (ILLIG *et al.*, 2011; SONG *et al.*, 2011) have made a distinction between **resource-centered user-centered approaches**. This refers to the fact that some approaches take user information into account when performing tag prediction (user-centered) while others take a global view, giving the same predictions for every user (resource-centered). Note that other authors, (*e.g.* ZHANG *et al.* (2014) and HU *et al.* (2010) refer to these two types as *personalized* and *collective* tag recommendation). The following table summarizes these differences:

Table 1.2: Tag prediction approaches, classified according to the information they use

Resource-centered approaches	Only information regarding the resources themselves is used to build the tag prediction mechanism. For a given resource, the same predictions are displayed for every user.
User-centered approaches	In addition to information about the resources, user-specific data (*e.g.* users' tagging history and profile) is leveraged for suggesting tags to be used at tag assignment time.

In this work, have chosen to limit our scope to **resource-centered approaches to predicting tags in broad folksonomies**. This is due to the previous reasons and to the fact that user-centered approaches do not perform so well vis-a-vis resource-centered methods (SONG *et al.*, 2011); the distribution of users and tags in broad folksonomies follow a power law and reusability of tags by each individual user is low.

Resource-centered approaches more robust as we generally have much more information about resources than about users. This is particularly true with textual resources and broad folksonomies. Also, resource-centered methods have the added benefit of being able to work in the absence of user information, the so called *cold start* problem.

More specifically, with regards to the chosen scope, we would like to inquire into the performance of different tag prediction techniques, as applied to different problem sets. We would like to be able able to answer questions such as:

- How do different methods of multi-label ranking perform when applied to the same data?
- How do dataset characteristics such as the total number of resources, average number of tags per resource, etc, affect the outcomes?
- Does the type of feature representation used affect the outcomes for different methods? If so, how?

In parallel to this, we would also like to investigate the effect of the sparsity of features on a specific label ranking method, namely *multi-instance multi-label SVM* (MIMLSVM), when applied to social tag prediction[3].

This is an interesting method that was originally used for scene classification (ZHANG & ZHOU, 2007). However, it was recently adapted for text classification, with satisfactory results. (SHEN *et al.*, 2009)

Given that there has been some work done on *representation learning*[4] for text (BENGIO *et al.*, 2003; LE & MIKOLOV, 2014; MIKOLOV *et al.*, 2013b) recently, we would like to investigate to what extent this particular method works when exposed to other kinds of text representations, namely *dense* representations.

For this, we would like to answer questions such as:

- Does MIMLSVM only work for the commonly-used bag-of-words representation?
- Do different types of dense representation affect the algorithm in different ways?
- Does the domain of the folksonomy under research affect the outcomes? If so, how?

1.3.1 Binary Tag Assignment Model

We would like to point out that most works in the literature do not take into account the number of times each tag has been assigned to a given resource. In other words,

[3]For detailed information on this, see section 5.2.

[4]*Representation Learning* (also called *Feature Learning*) refers to using machine learning methods just to find good representations of data. In other words, the objective of these unsupervised methods is to learn useful ways to represent instances. These can then be used in traditional supervised classifiers or used as is.

a *binary* tag-assignment model or *BTAS* (ILLIG *et al.*, 2011) is used, whereby a tag assignment is equated with the fact that *there exists at least one* user who assigned that tag to that resource. We follow that convention in this article.

Definition. BTAS[5]

Let TAS be all tag assignments made by all users $u \in U$, using tags $t \in T$ for resources $r \in R$:

$$\text{TAS} = \{(u, t, r) \in U \times T \times R \mid \text{user } u \text{ has assigned tag } t \text{ to resource } r\}$$

BTAS abstracts the user dimension away, considering instead a *binary* tag assignment (t, r) the existence of *any* user u having assigned tag t to resource r:

$$\text{BTAS} = \{(t, r) \in T \times R \mid \exists u \in U : (u, t, r) \in \text{TAS}\}$$

More information on these issues is given in chapter 2.

1.4 Methodology

Here we will give a brief overview of the research methodology used in this work.

1.4.1 Literature Review

In order to add replicability and transparency to our literature review, we partially adopted principles from Systematic Literature Review, as defined in works such as BAUMEISTER & LEARY (1997), WOHLIN (2014) and BEM (1995).[6]

We selected three reputable repositories of scientific articles and research pieces, namely IEEE-XPlore Digital Library[7], ScienceDirect[8] and Scopus[9].

After initial contact with the subject matter of our work, we selected four sets of search terms, namely *"collaborative tagging"*, *"social tag prediction"*, *"social tagging"* and *"tag prediction"* and used those to search the titles, abstracts and contents of research pieces in the websites' databases.[10]

We gathered and organized the results of the aforementioned queries; after removing duplicated entries, we had a collection of 2466 articles, book chapters or

[5] Adapted from ILLIG *et al.* (2011).

[6] Evidence such as screenshots of search results and the actual set of articles retrieved from each query can be provided upon request.

[7] http://ieeexplore.ieee.org/Xplore/home.jsp

[8] https://www.sciencedirect.com/

[9] https://www.scopus.com/

[10] These search terms were chosen because we believe they encompass a large part of the available literature related to the topic of our work which we defined (for the purpose of this literature review) as **"Predicting or recommending tags in a social tagging environment"**.

conference proceedings. No date filter was used, and only texts in English were selected.

We read the abstracts of all 2466 pieces and, based on that, we selected 399 as being somehow related to the subject of our work, as explained above.

Out of these 399 relevant works, we further refined our set to 285 articles, by extending our analysis to the introduction and conclusion sections. This final list of 285 articles all contained information directly related to the task of predicting and/or recommending tags in a social tagging environment. They were all read in order to inform our research, although not all were included or cited in this text.[11] Naturally, many articles not on this list were also read because they were referenced very often. In this light, this formal method of searching for articles only produces an initial list of articles to read; citations obviously lead us to other works eventually.

1.5 Document structure

In this chapter we introduced the subject matter for this dissertation and the work methodology we will use.

In Chapter 2 we will give a brief overview of Social Tagging Systems and Folksonomies. In Chapter 3, related work is analyzed and compared. In Chapter 4 we will propose a solutions to the research questions we previously highlighted, and describe the way we intend to address them. In Chapter 5 we describe the experiments conducted and analyze the results. Finally, in Chapter 6 we conclude this dissertation and provide pointers for future work and ways in which it can be extended and/or continued.

[11] Many other articles, which did not feature in the search results, but were obviously relevant (based upon citation count for example), were also added and read.

Chapter 2

Social Tagging

Continuing the description of Social Tagging we started in the introduction, we will now go into more detail about this concept, as well as some related terms (such as folksonomies) in the next sections.

2.1 Examples of Social Tagging Systems

Examples of STSs abound in the modern Web. We will present two different examples so that the reader can better grasp what a Social Tagging System looks like in practice.

2.1.1 MovieLens

MovieLens[1] is a research website run by GroupLens Research at the University of Minessota.

It provides users with personalized movie recommendations based on how they have rated individual films. In addition to information and ratings for many movies, MovieLens also lets users add tags to movies and view tags others have assigned.

As can be seen in the following image, tags allow users to give objective (*car chase, espionage*) and subjective (*great plot* attributes to resources, in this case films.

[1] https://movielens.org

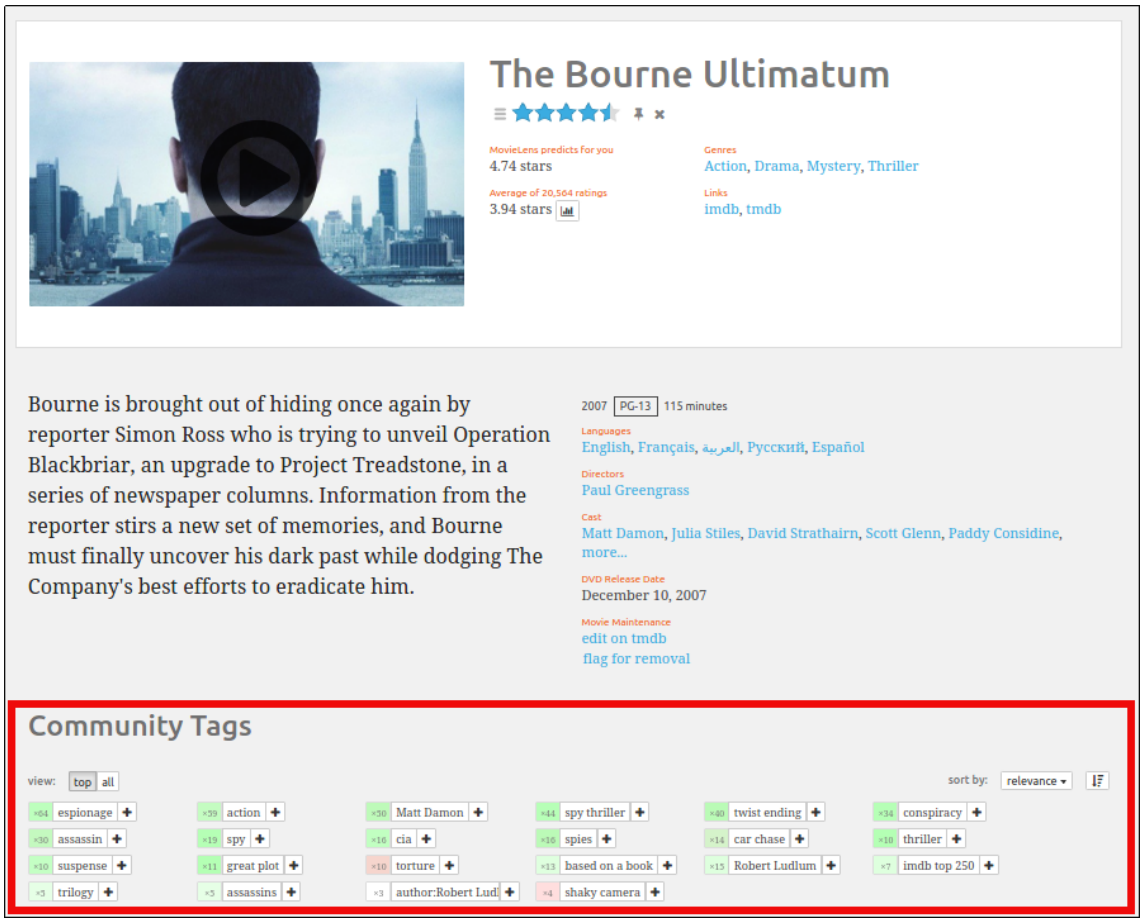


Figure 2.1: The MovieLens website supports tagging; any user can add their own tags and view tags assigned by other users to a particular resource. Retrieved from https://movielens.org/movies/54286 in January 2018.

2.1.2 StackOverflow

StackOverflow[2] is a very popular Q&A (Question and Answer) website. It receives roughly 8,000 new questions related to computer programming every day.[3]

StackOverflow supports tagging of questions; users can add up to 5 tags to every question they post. Among other features, tags can be used to narrow down search results and they can also be subscribed to. Tags are also part of the website's incentive and reputation mechanisms; you can be awarded *tag medals* for completing specific objectives such as answer many questions having a particular tag.

[2]https://stackoverflow.com/
[3]As of 2017: https://stackoverflow.blog/2017/05/09/introducing-stack-overflow-trends/

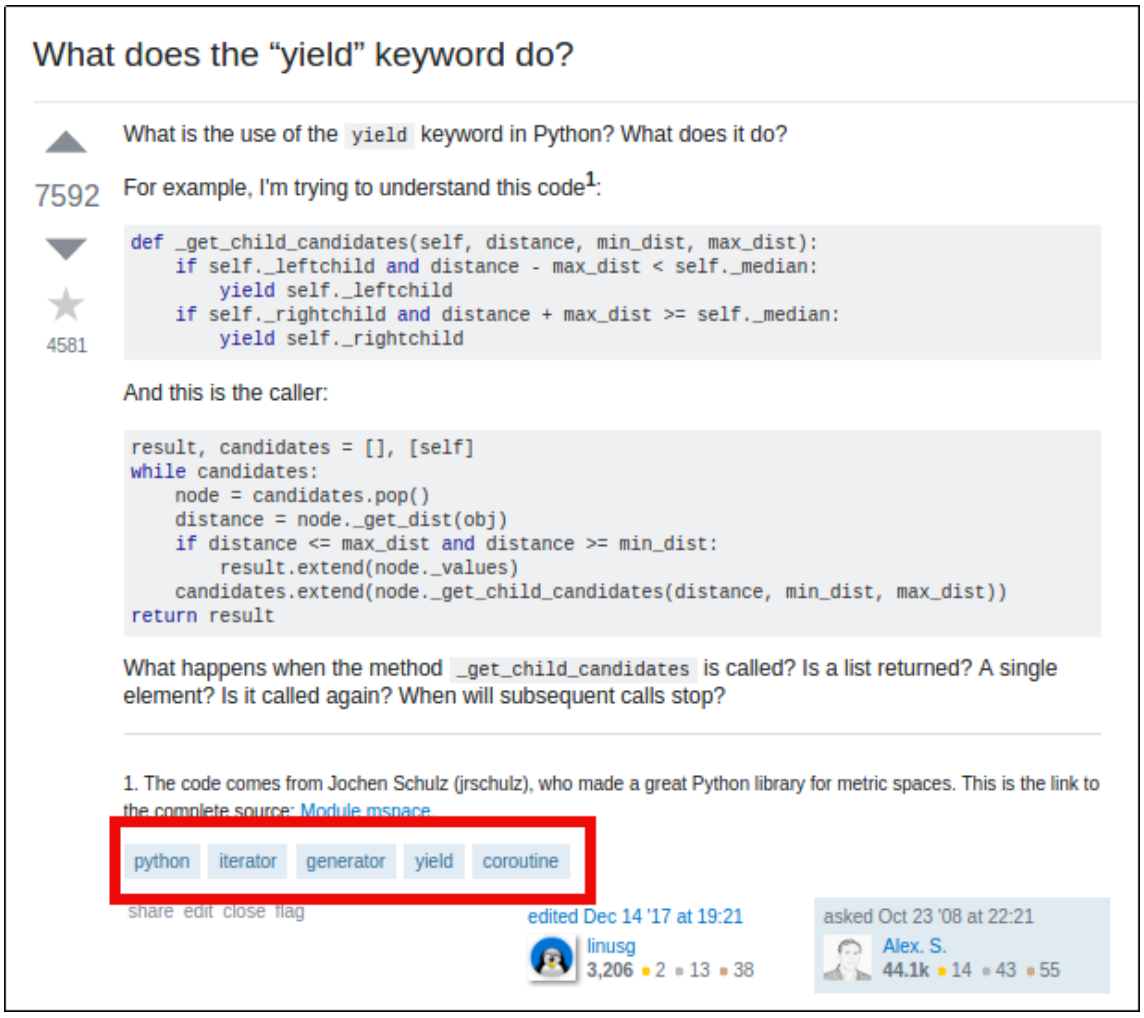


Figure 2.2: The StackOverflow website also supports tagging, but only a single set of tags is shown, namely the tags assigned by the resource's original owner (and maybe edited afterwards). Retrieved from https://stackoverflow.com/questions/231767/what-does-the-yield-keyword-do in January 2018.

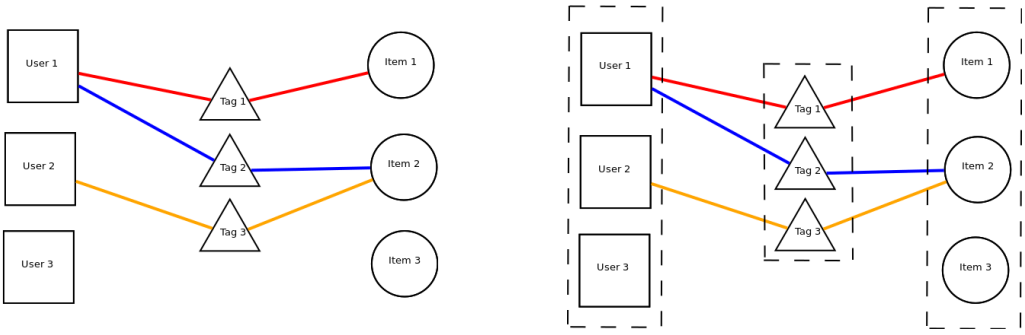
Figure 2.3: Tags are also used to help drive Stackoverflow's incentive mechanisms; tag medals are awarded for activity related to a certain tag. Retrieved in January 2018. (Blur is used to protect the user's privacy)

2.2 Social Tagging and Folksonomies

Since the beginning of STSs, it has been observed that such systems grow in an organic way and that certain patterns are noticed with respect to how tags are used. As an example, it has been observed (HALPIN *et al.*, 2006) that the number of times each tag is used to tag a particular resource forms a *power law*, i.e. some tags are used exponentially more often than others.

More generally, the term **folksonomy** has been used to describe these emerging patterns of informal organization and meaning assumed by tags in a Social Tagging System (MATHES, 2004; WAL, 2005b).

According to MIKA (2007), one way to model folksonomies is via *tripartite hypergraphs*. Hypergraphs are generalizations of graphs (BERGE, 1985) where edges can join not just two but multiple nodes. Furthermore, hypergraphs representing folksonomies are also tripartite, inasmuch as there is a three-way *partitioning* scheme (namely users-resources-tags) such that edges do not connect nodes that are in the same partition:



(a) User1-Tag1-Item1 (in red) is a single *hyperedge* in this folksonomy hypergraph.

(b) This is a *tripartite* hypergraph because we can find 3 disjoint partitions

Figure 2.4: A folksonomy can be represented as a tripartite hypergraph, where three-way hyperedges connect users, tags and items. Adapted from RAWASHDEH *et al.* (2013). (Best viewed in colour).

The word *folksonomy* itself (formed by *folk* + *taxonomy*) points to the fact that, differently from a rigid, often expert-driven taxonomy, the patterns that arise with the free use of tags by a community follows a more fluid, hapzard fashion, as can be visualized in the next image where the the tag *afghanistan* was used as search criterium on a photo-sharing website; searching for images tagged *afghanistan* on Flickr yields pictures from the Afghani people, the war in Afghanistan, the Afghani landscape, etc. This reflects the multitude of meanings a single tag may acquire due to the way users tag their pictures.

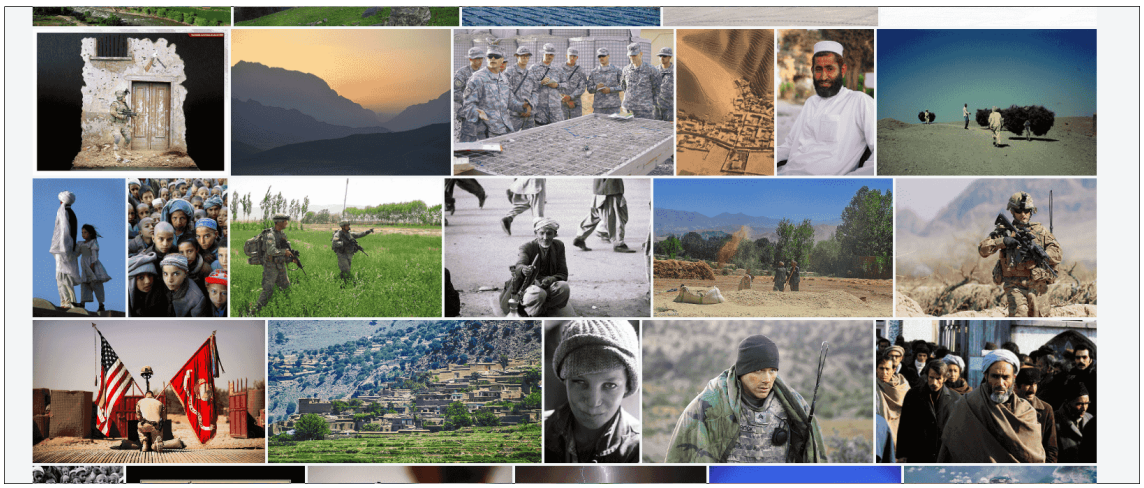


Figure 2.5: Retrieved from https://www.flickr.com/search/?tags=afghanistan in January 2018.

2.3 Narrow and Broad Folksonomies

Based on the examples given and one's general day-to-day experience, it's natural to conclude that folksonomies and their underlying STSs vary widely with respect to their features and how they're implemented.

One basic difference, raised as early as 2005 (WAL, 2005a) and commented on by other authors (HALPIN *et al.*, 2006; MARLOW *et al.*, 2006; PETERS, 2009) since, is that between *narrow* and *broad* folksonomies.

As mentioned before, **narrow** folksonomies are folksonomies which only allow a resource's original poster (commonly referred to as *O.P.* in such systems) to add tags to that resource. In other words, *users can only tag their own content.* Conversely, **broad** folksonomies are those where every user can add tags to any resource on the platform.

This distinction is relevant for researchers studying the dynamics of social tagging systems. For example, it has been noted by SCHIFANELLA *et al.* (2010) that a global, shared tag vocabulary cannot be observed in narrow folksonomies, unless it is specifically promoted by the system.

Broad folksonomies exhibit more diversity and richness of information, not to mention sheer scale, which makes them more amenable to analysis by data-driven methods, such as machine learning. More concretely, it has been suggested that a shared, global vocabulary of tags cannot be observed in narrow folksonomies (SCHIFANELLA *et al.*, 2010).

On a similar note, AIELLO *et al.* (2012) have suggested that tag predicting is more meaningful in broad folksonomies, since users can tag the same, global, set of resources. Also, tagging in such systems tend to reflect resource contents rather than users' personal preferences.

As related to the ease of navigation in STSs, HELIC *et al.* (2012) have suggested that broad folksonomies are better and more efficient for user browsing, inasmuch as these tags encode more information than their counterparts in narrow systems.

2.4 Other Aspects

Here we will talk about a few other aspects which we deem relevant in light of this work's objective, namely that of predicting tags in STSs.

2.4.1 Tag Stabilization and Convergence

In broad folksonomies (*i.e.* those where all users can add tags to any resource), the tag distribution for a given resource has been observed to stabilize after about 100 individual tag assignments (GOLDER & HUBERMAN, 2006). Reasons given for

this phenomenon include *imitation*, *i.e.* users are influenced by other tags already given to a resource and *shared knowledge*, i.e. other tags help build a user's mental model of the meaning for each tag. We consider this an important result because it may affect the level of tag prediction we can achieve.

One can attest to this phenomenon in the following image; it clearly shows that once that critical level is reached, the tag distribution for a given resource hardly changes anymore.

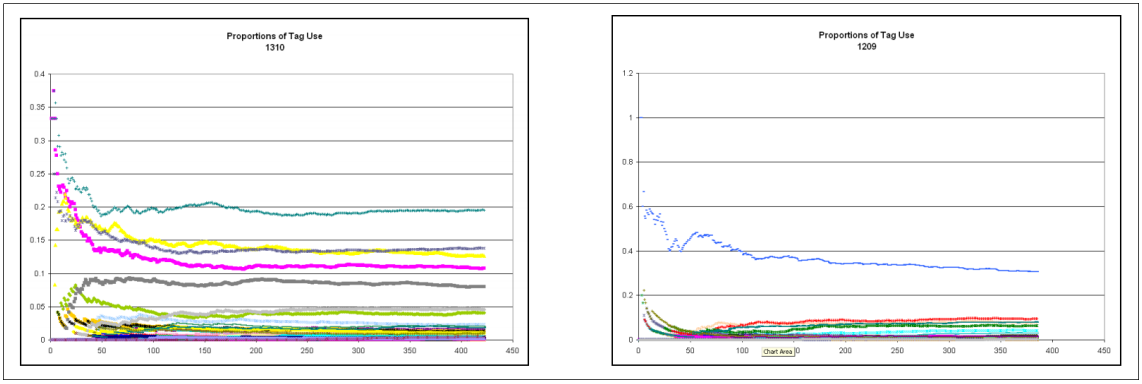


Figure 2.6: Tag distributions for two resources on Delicious.com. Tag proportions reach equilibrium after around 100 tag assignments. Adapted from GOLDER & HUBERMAN (2005)

2.4.2 Effect of tag suggestion on STSs

It has been suggested by MARLOW *et al.* (2006) that a STS falls under one of three types depending upon how much system support there is for tagging:

- *Blind Tagging*: Users cannot view other tags assigned to an item, before adding their own.
- *Viewable Tagging*: Users can view other tags assigned to an item before adding their own tags.
- *Suggestive Tagging*: Users can not only view other tags but the system also suggests appropriate tags.(MARLOW *et al.*, 2006)

They have suggested that the level of tagging support (as referred to above) present in a system may make tag stabilization and convergence faster.

In light of that, we can suggest that tag prediction can contribute to a higher quality STS, if we assume that a folksonomy where the global vocabulary has converged is more useful than one where it hasn't.

Chapter 3

Related Work

3.1 Introduction

In this section, we will present general approaches to tag prediction, with a special focus on resource-based methods for broad folksonomies, that being our problem scope.

After reviewing the literature, we chose to divide these methods in categories for easier analysis:

- **Association Rule Mining**: Methods that leverage the learning of empirical co-occurrence rules in the datasets.
- **Content-based Tag Propagation**: Methods that learn a representation of each resource based on their contents and use neighbour-based techniques to find similar points.
- **Resource-based tag propagation**: Similar to the above, but using other information to build representations for each resource.
- **Multi-label Classification/Ranking**: Methods based upon training multi-label classification algorithms, ranked or otherwise.
- **Topic Modelling/Tensor Factorization**: Methods based on finding a matrix and/or tensor based representation for resources and tags, and then applying factorization methods on those.
- **Graph-based**: Methods which model folksonomies as graphs and leverage graph-theoretic algorithms to predict tags for resources.
- **Other**: Other methods not in the previous categories.

As related to how authors name their particular approaches, one should be careful inasmuch as there is no apparent consensus as to what constitutes a *recommendation* approach vis-a-vis a *prediction* approach with respect to STSs. It is frequently the case in the literature that the word "recommendation" is used to refer to methods that use no user-specific information whatsoever and, conversely, words like "prediction" and "suggestion" used in cases where personalized recommendations are made.

3.2 Resource-centered Methods

In this section, we present a collection of *resource-centered* methods for tag prediction. They are so called because they leverage only resource-specific information in order to predict what tags should be assigned to a new, unlabelled resource. In other words, all prediction are of *unpersonalized* nature.

We note that, although our problem domain only includes textual documents, we chose to also mention in this section approaches used in other domains, such as audio, video and images. This is because we are mostly interested in how these approaches work irrespective of the choice of features used.

3.2.1 Association Rule Mining

Association rule mining[1] refer to methods whereby one extracts rules and regularities from event databases (AGRAWAL *et al.*, 1993). For example, the rule {Beer,Bread} $\rightarrow$ {Milk}, when referring to a database of supermarket items, may indicate that Beer and Bread jointly *co-occur* very frequently with Milk, suggesting a possible relationship between the two.

In one of the earlier papers on social tag prediction, HEYMANN et al. (2008) have applied *association rules* of the form $X \rightarrow Y$ (where X and Y are tagsets) in order to expand the set of tags given to a resource. Using techniques such as *Market Basket Analysis*, the authors derive association rules of length 4 and below, using a certain level of support[2] as threshold to remove overly noisy rules.

The authors report (HEYMANN *et al.*, 2008) that a surprisingly high number of high quality rules can be found (such as those representing *type-of* relationships and synonyms). Furthermore, the added tags help increase precision and recall for user queries, when the result sets are augmented to include documents tagged with those tags.

[1] Alternatively, *Association rule learning*.

[2] A rule's *support* equals the number of examples where both X and Y are present.

The authors also claim that using larger and larger rules would probably increase performance, but computational complexity quickly become prohibitive.[3]

Another approach involving association rules was put forward by VAN LEEUWEN & PUSPITANINGRUM (2012). They acknowledge the fact that gains in performance brought about by using larger rules come at a high cost in terms of processing time. They, however, suggest that a compromise can be achieved by choosing a carefully selected set of association rules, such that performance is increased at a lower cost.

Their approach works by using a compression mechanism to efficiently compute expanded tagsets for any given tagset. It computes the most suitable expanded tagsets ranked by support.[4]

3.2.2 Content-based tag propagation

Here we provide a basic overview of methods which, in one way or another, use the content-based similarity to propagate tags from labelled instances to unlabelled ones.

In the first approach, SORDO *et al.* (2007) have used both first-order (stylistic) and second-order (mood-based, extracted from the stylistic ones) features and a neighbours-based similarity measure to propagate labels from labelled audio pieces to unlabelled ones.

They reported good results for the approach, as measured by rank-based metrics such as Spearman's rank and Precision@k. They claim that ignoring tags with too few assignments improves results and that sometimes using more neighbors is beneficial, while sometimes it's harmful.

It should be noted here that this work was not run on a broad folksonomy, since all examples were annotated by a single person.

Another interesting example is that of MOXLEY *et al.* (2008). Their approach uses many feature *modalities* to represent a resource (in this case, videos). In other words, they use multiple sources of information to build a feature vector, namely text information from the video transcripts, image information from video snapshots and concept information from external source.

They report good results using set-based performance metrics (slight variants of precision and recall). Furthermore, they claim that using an average of features built from multiple modalities helps suppress the effect of noisy information.

[3]Note that this method assumes that a resource already has some tags assigned to it. These are then used to predict another set of tags. This is sometimes called **tag-set expansion** in order to differentiate it from methods that do not make this assumption.

[4]The support for a given tagset is simply the number of times that particular tagset was assigned to a resource in the system.

In GUILLAUMIN *et al.* (2009), the authors propose a weighted neighbor approach where one can choose an arbitrary distance measure (i.e. Euclidean, Manhattan, etc) one wishes to use to measure similarity between resource representations. Then, the optimal weights for each resource are found via the optimization of a custom loss function that encodes the accuracy each individual tag prediction.

In other words, the dataset is used to inform the decision on what weights to use for each resource. This will, in turn, define to what extent tag assignments for each resource will influence those of its neighbors.

This approach has been called *metric learning* and, according to the authors, it has been used in the past in other contexts.

In LI *et al.* (2009), the authors have approached the problem from a slightly different angle. Although they have also used content-based similarity to search for neighbors, the weight given to each tag is not just proportional to the similarity between each pair of neighbors; it also incorporates a term that normalizes each tag according to the tag's *prior*, i.e. the overall frequency of a given tag in the whole dataset.

By using rank-based metrics such as Precision@k and Mean Average Precision (MAP), they report that their method consistently outperforms approaches that do not take a tag's overall prior into account.

In conclusion, two common themes in such *content-based* tag propagation approaches seem to be **a)** designing similarity measures and other ways to retrieve similar resources given a query resource and **b)** once the neighbor resources are found, find meaningful ways to weigh the contribution given by each neighbor in order to predict tags for the query resource.

3.2.3 Resource-based tag propagation

In this subsection, we will talk about methods which use information *about* the resource (other than its contents encoded as features) to build representations for these resources. These representations are then used in neighbor-based algorithms for actual classification.

AU YEUNG *et al.* (2009) propose a slightly different approach. They encode each resource as a vector over the space of the full tag vocabulary, so that it resembles a bag-of-words approach, using tags instead of terms in the document. Similarity between resources is then calculated via simple measures like cosine similarity.

The authors report above-benchmark performance when using the described approach to predict tags for unlabelled examples. Metrics used to for gauging performance include Precision@k and NDCG.

3.2.4 Multi-label Classification/Ranking

Since resources in an STS can be assigned multiple tags, it is natural to model this problem as a *Multi-label Classification* (MLC) problem.

Multi-label learning[5] (TSOUMAKAS & KATAKIS (2007)) refers to learning from data that is multi-labelled, that is, data where each example has not just a single label[6] but multiple ones.

A more particular approach, generally called *Multi-label Ranking*, refers (ILLIG *et al.* (2011)) to problems where not only do instances have multiple labels associated with them, but every label also has a *rank*; in other words, each label assignment also carries a weight, so that labels assigned to a particular example may be ranked with respect to the weight each label has. This is in contrast with regular multi-label classification, where labels are represented with a binary vector, making no distinction between labels.[7]

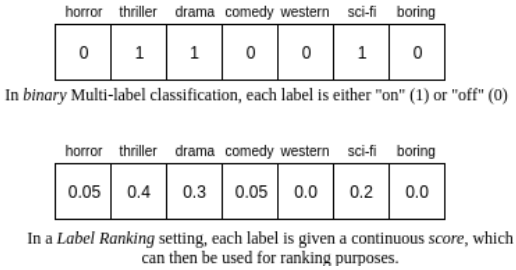


Figure 3.1: When each label prediction is given a score, we can choose a threshold k and return only the top k labels, as ranked by score.

In KATAKIS *et al.* (2008), the authors have applied multi-label classification to the task of classifying HTML pages and journal abstracts into tags.

The chosen method was to train a binary classifier for each individual label, a meta-classification procedure called *Binary Relevance* [8] in the MLC community. The underlying classifier was a simple Naïve Bayes model, trained on the bag-of-words representation of the text documents.

The authors claim good results with their model, while noting that they have restricted the tag vocabulary to those tags appearing in at least 50 documents in order to trim rare tags.

BERTIN-MAHIEUX *et al.* (2008) use a 2-level model to predict tags for audio pieces from a popular STS for songs, namely *last.fm*[9].

[5]Not to be confused with *multi-class* classification.

[6]Problems where each example has a single label are, unsurprisingly, referred to as *single-label* classification in MLC literature

[7]Although our own method is of the multi-label ranking type, we find it worthwhile to list regular MLC method due to how similar both are.

[8]*Binary Relevance* is an adaptation of the well-known *One-versus-All* (RIFKIN & KLAUTAU, 2004) classifier, commonly used for multi-class classification.

[9]Reachable online via http://last.fm

For the first level, they use a technique called *Filter Boosting*, which is an extension to *Adaptive Boosting* meta-learning, better suited at online learning with large datasets. Using decision stumps (decision trees with a single level) as the underlying classifier, they train an individual classifier for each tag, which is also used to extract features to be used downstream.

The second level is another Filter Boosting classifier trained on the output of the first one, possibly dropping features found to be irrelevant by the first level.

They reportedly beat previous performances on this particular dataset and noted that using the first level for feature selection seems to help with generalization.

SHEN *et al.* (2009) introduce a different approach to predicting tags, namely one leveraging *multi-instance* learning (DIETTERICH *et al.*, 1997), whereby one considers a single training example as a *bag* of instances, rather than a single entity.

The technique[10] combines multi-instance learning with multi-label learning by splitting a single resource (in this case, tagged documents from the *Delicious* website) into a bag of individual parts, combining these into a single instance by means of clustering and then using those for classifying the original resource into multiple tags.

More specifically, they use a well-known text segmentation algorithm called *TextTiling* (HEARST, 1994) to split each document into segments. This turns each document into a bag of segments. Then, as per the technique, each bag of segments is transformed into a single feature vector, by means of *k-medoids* clustering (KAUFMAN & ROUSSEEUW, 1987).[11] Once the problem has been reduced into a regular multi-label ranking problem, a simple *one-vs-all* metaclassifier using an SVM model is used for actually predicting tag scores.

The authors report that this method compares favourably against other common multi-label models such as Binary Relevance and ML-k-NN (ZHANG & ZHOU, 2007), as evaluated by metrics such as Precision @k, Recall @k and Accuracy @k.

SONG *et al.* (2011) is an interesting work inasmuch as almost equal attention is given to performance and to training time. They train an adapted *Gaussian Process* model on three different datasets with varying characteristics (*Delicious, Bibsonomy and CiteULike*). They argue that Gaussian Processes are a good fit to the problem at hand (label ranking) because they naturally output posterior probabilities for each class, which can be naturally used for label ranking.

In order to make training and inference faster, they only choose M, where $M <<< N$ to estimate the hyperparameters for the model, yielding significant gains in training and test times.

[10]This technique was adapted from a previous work by ZHANG & ZHOU (2006) in scene classification

[11]In order to enable clustering of multiple bags of vectors, a custom distance metric needs to be used. In this case, the *Hausdorff* distance metric (HUTTENLOCHER *et al.*, 1993) was used.

Notably, the authors report that their method outperforms competitive alternatives such as SVM by as much as 30% while only using 5% of the training data (due to the selection of prototypes).

KATARIA & AGARWAL (2015a) have leveraged recent research on word and document-level embeddings (LE & MIKOLOV, 2014; MIKOLOV *et al.*, 2013) to build text representations specifically for tag prediction in the context of an STS. Their model, named *Tags2Vec*, extends the *ParagraphVector* framework by using tag assignment information in addition to word contexts.

The original *ParagraphVector* model uses a shallow neural network in an unsupervised way to induce document and word representations, by using an objective function that forces a document to be a good predictor or words that occur in it. *Tags2Vec* augments the objective function so that, in addition to words, a document's representation should be also good at predicting tags that are assigned to it.

These document representations were then used to train SVM and Gaussian Process classifiers using two datasets: *CiteULike* and *Delicious*. The models trained using *Tags2Vec* representations significantly outperform analogous models trained on other representations such as *ParagraphVector* and the traditional TF-IDF vectors, indicating that the additional tag information has indeed helped in inducing better representations for documents in an STS setting.

TAO & YAO (2016) also made use of *ParagraphVector* to represent documents in a Chinese STS, namely *ZhiHu*[12]. These representations were then used to train a One-versus-Rest SVM classifier and also a neural network with one output node for each tag.[13]

They report better results when using document embeddings vis-a-vis bag-of-words features. In addition, they report that results using One-vs-Rest SVM are also better than those obtained using neural networks.

This is an interesting example because it shows the relative performance of neural networks with respect to SVM classifiers. It also shows that document embeddings work for the Chinese language, which has very different structure and syntax when compared to western languages.

[12]https://www.zhihu.com

[13]This is a commonly-used way to train neural nets for multi-label problems. While normal neural nets use softmax activations on the last layer, it's also possible to use N output nodes (where N is the size of the tag vocabulary) to obtain individual predictions for each tag.

3.2.5 Methods based on Topic Modelling/Tensor Factorization

We now turn our attention to methods that leverage *Topic Modelling* and/or Tensor Factorization. We group these two topics together because topic modelling and tensor factorization are sometimes intimately related, e.g. *Latent Semantic Analysis* (LSA) (DEERWESTER *et al.*, 1990) is nothing but Singular Value Decomposition (SVD) applied to a term-document matrix.

Topic modelling methods used include LSA and variations (ZHANG *et al.*, 2014) and LDA and variations (GONG *et al.*, 2017; SI & SUN, 2008; WU *et al.*, 2016).

With regards to tensor factorization, this method is heavily used in *user-centered* approaches such as RENDLE & SCHMIDT-THIEME (2009), RENDLE *et al.* (2009) and SYMEONIDIS *et al.* (2008), all of whom model the user-resource-tag relation as a tensor, and apply factorization to arrive at more economic representations that can be used for predicting unlabelled resources.

Latent Dirichlet Allocation (LDA) (BLEI *et al.*, 2003) is a well-known Topic Modelling method for learning the best way to represent a given corpus into topics. On broad lines, LDA models each document as a distribution over topics which, in turn, are distribution over words.[14] As the model is generally intractable, one uses methods such as variational inference (as in the original paper itself) of MCMC-based methods such as Gibbs sampling.

Tag-LDA is a method introduced by SI & SUN (2008)[15], which extends LDA to account for tags in addition to words in a document. In other words, a model is trained to find topics which are not only distributions over words (as in the original LDA model) but distributions over *words and tags*.

Since this is a supervised model aimed at predicting tags for unseen documents, the test time procedure is as follows: the most likely topic distribution for the query document are calculated, and the most likely tags for each of the topics are retrieved:

$$p(t|d) = \sum_{z \in Z_d} p(t|z) \cdot p(z|d) \ , \tag{3.1}$$

where Z_d is the set of topics assigned to document d at test time.

KRESTEL & FANKHAUSER (2010) also leverage LDA for predicting tags but they use a different approach. They use no content information but just a resource's

[14]The Dirichlet distribution can be seen as a distribution over the space of possible parameter vectors for a multinomial distribution.

[15]Other authors such as HU *et al.* (2012b) have created slight variations on this method.

previous tag assignments as its representation.[16] In other words, instead of documents composed of terms, this approach models resources composed of tags.

At training time, hyperparameters for the Dirichlet distribution are inferred, so that a certain number of tag *topics* are found. Each topic is a vector of probabilities for each tag in the vocabulary. At prediction time, the most likely topics for a given resource are estimated and the most likely tags are predicted for that resource. This is similar to the previous work by SI & SUN (2008).

The authors note that the performance of this method is not as good as when using regular LDA on documents and terms, because the number of tags assigned to a resource is orders of magnitude smaller than the usual number of terms in a document, making it harder for LDA to correctly infer good topics (KRESTEL & FANKHAUSER, 2010).

In the article ZHANG *et al.* (2014), the authors also use a topic modelling approach, namely a modified version of Latent Semantic Analysis (LSA) (DEERWESTER *et al.*, 1990), applied on the resource-tag matrix. They add an additional constraint to LSA, by requiring that all elements in the reduced matrix be *nonnegative*.[17] The nonnegativity constraint helps with interterpretability and ensures the factor matrices are sparse (GILLIS, 2014).

At training time, the LSA model is trained on the training set (the resource-tag matrix). At inference time, a query resource is projected from the resource-tag space to the topic space, yielding topic probabilities. Finally, tags are suggested for the new resource using the same approach as SI & SUN (2008).

They report that their method outperforms similar topic-modelling and/or dimensionality reduction approaches, such as SVD, LDA and k-NN.

3.2.6 Graph-based

It is usually the case that a folksonomy is modelled as a tripartite graph, as explained on section 2.2. Many methods take advantage of that fact to leverage graph-based algorithms such as PageRank 5

These methods generally model folksonomies in terms of a graph $G = \langle V, E \rangle$, where V is the set of nodes representing resources, E is the set of edges, which connects nodes if they share a common tag.

Although the methods we describe next all take a resource-centered approach to tag prediction[18], we deem worthwhile to mention that it is in *user-centered* approaches that graph-based models have been more heavily used. The most widely used and cited graph-based method is probably *FolkRank* (JÄSCHKE *et al.*, 2007),

[16] Note that this method assumes that a resource has been assigned at least one tag already.
[17] Such methods are generally called Nonnegative Matrix Factorization (NMF).
[18] Because this dissertation is focused on this type of methods.

an adaptation of the famous *PageRank* algorithm (PAGE *et al.*, 1999), trained to predict tags in a personalized manner. Methods based on Random Walks are also commonly used in user-centered approaches (JIN *et al.*, 2010; MROSEK *et al.*, 2009; SI *et al.*, 2009).

WANG *et al.* (2015) models resources (in this case, web pages) and tags as a graph $G = \langle V, E, \omega \rangle$, where ω is a function that defines the weight of the connection between two nodes. Function ω assigns a weight between two nodes such that it is larger if the two nodes' textual contents is similar, and smaller otherwise.

Once this modelling is complete, the authors apply a clustering method called *DenShrink* (HUANG *et al.*, 2011) which clusters nodes together. At prediction time, one uses tags in the same cluster to suggest for resources with few or no tags. The authors claim that this method succeeds at suggesting tags for what they call *hesitant* (i.e. with few or no assigned tags) but they do not compare it to other methods in the literature.

KAKADE & KAKADE (2013) propose a different graph-based approach, wherein nodes are resources (in this case, images) and tags. However, differently from previous methods, they build 3 different graphs: in the first graph edges connect resources to tags. In the second graph, nodes are resources and egdes connect resources to other resources, based on feature similarity. Finally, in the third graph, tags are connected to other tags, based on how often they occur together. They call this a *fused graph*.

At prediction time, they perform a *random walk* in these graphs; using the query resources features, they find similar images on the image-image graph. Then, they apply the same method on the image-tag graph and then on the tag-tag graph, in order to arrive at a set of suggested tags for the query resource. They compare multiple variations of their algorithm and conclude that the best performance is achieved using all three graphs and, in addition, performing a technique called *Pseudo Relevance Feedback*.

3.2.7 Other

In this subsection, we describe a few more methods which do not fit into the previous categories. However, we nonetheless deem them important because they show that methods applied to predicting social tags are not limited to the ones in the categories previously mentioned.

SI & SUN (2010) introduce a generative probabilistic model aimed at inferring the latent *reasons* behind every tag assignment. This is somewhat inspired by LDA but they also model the *noise* inherent in all STSs.[19]

[19]This is especially true of *broad* STSs, because of the fact that all users can tag all resources.

The authors claim that their model outperforms baseline methods like k-NN and Naïve Bayes on multiple datasets.

TRABELSI *et al.* (2012) employ Hidden Markov Models (HMM) (RABINER, 1990) to build prediction model for tags. At training time, They model the sequences of users' latent (hidden) *intents* when tagging a resource as the hidden states in the HMM, and the actual tag assignments are the visible states.

At prediction time, the model is used in reverse to infer the hidden state from the observable data. Finally, tags related to the most likely hidden *intent* are suggested for the query resource. They claim their method outperforms similar probabilistic methods in terms of prediction and recall, for multiple values of k.

With an ensemble-based approach, LIU *et al.* (2013) propose a *blending* of multiple method into a single classifier.

It works as follows: they first extract features from the resources and then train three individual classifiers[20], namely a simple keyword extractor, item-based collaborative filtering and LDA. Next, they train a linear model to find out what are the optimal weights λ such that a linear combination of the results of the three individual is better than each individual classifier.

In order to ascertain the performance of the proposed solution, they compare the output to each of the individual underlying classifiers, and conclude that the blending method succeeds at increasing performance on a crawl of the *Delicious* website.

SATTIGERI *et al.* (2014) propose using Deep Architectures for learning good features for audio tagging. More precisely, they train low-dimensional representations (*embeddings*) for audio data, apply a sparse transformation and then cluster the obtained features into similar categories. Finally, they apply a simple Linear SVM classifier on the final features. The authors claim that their approach has comparable performance to the best competitors at the time, even though a relatively simple classifier was used.

Although this method is specifically used for the audio domain, we believe that it highlights that the feature extraction step may be just as important as choices over which classifiers and hyperparameters to use.

The following table provides a quick summary of the main types of approaches described in this section:

[20]They cite general ensembling theory, whereby an ensemble of unrelated, non-correlated weak classifiers works better than combining classifiers which are similar to one another.

Table 3.1: Approaches to tag prediction, classified by techniques used

Association Rule Mining	Methods that leverage the learning of empirical co-occurrence rules in the datasets.
Content-based Tag Propagation	Methods that learn a representation of each resource based on their contents and use neighbour-based techniques to find similar points.
Resource-based tag propagation	Similar to the above, but using other information to build representations for each resource.
Multi-label Classification/Ranking	Methods based upon training multi-label classification algorithms, ranked or otherwise.
Topic Modelling/Tensor Factorization	Methods based on finding a matrix and/or tensor based representation for resources and tags, and then applying factorization methods on those.
Graph-based	Methods which model folksonomies as graphs and leverage graph-theoretic algorithms to predict tags for resources.
Other	Other methods not in the previous categories.

3.3 Other Aspects

In this section, we will go over a couple of aspects we deem important inasmuch as they are *model-agnostic* - these can be use no matter what approach one takes for predicting tags in a social tagging context.

3.3.1 Data Representation

Feature representation is an essential part of any kind of machine learning, because any kind of information (be it text, images, sound, etc) must be encoded as vectors so that models can be trained on them. In this subsection we cite a couple of approaches that have leveraged alternative feature representations for the task of predicting social tags.

HAN *et al.* (2010) suggest an interesting technique wherein they use concepts from *transfer learning* (PAN & YANG, 2010) to train an embedding matrix M on a training dataset. At test time, M is used to project the test data into a lower dimensional vector space such that the correlations between the multiple labels are kept. Finally, a simple regularized linear regression model is used to train an independent classifier for each label.

The authors claim that their method outperforms similar and baseline methods on an image tagging dataset, as measured by ranked metrics such as Mean Average Precision (MAP) and Precision@k.

KATARIA & AGARWAL (2015a) suggest an approach whereby resources (in this case, text documents) and tags are represented in the same shared subspace. More specifically, they train so-called *translational embeddings* (BORDES *et al.*, 2013a) using a shallow neural network that forces the feature vectors to assume representations that minimize a loss function that represents the relationship between documents and tags (and optionally users).[21]

They claim their method outperfoms baseline methods, when used as preprocessing step for multi-label classification problems, where the actual classifiers may be neural networks, SVMs or method based on Gaussian Processes.

3.3.2 Clustering

Clustering[22] refers to a type of unsupervised machine learning techniques whose objective is to group instances in order to extract common patterns and other similarities.

SHEN *et al.* (2009)[23] use k-medoids clustering to predict tags for text data. They first break up each individual document into segments by using the *TextTiling* procedure (HEARST, 1994) and then cluster the segments back together. Finally, classification is done using SVM classifiers.

In a somewhat similar approach, NIKOLOPOULOS *et al.* (2009) use k-means with additional connectivity constraints to break up images into regions.[24] Then, regions from multiple images are clustered together in order to find the general topics represented in the images. Using a labelled dataset with multiple tags for each image, they construct a derived dataset where each region cluster is assigned the most common tags for all the regions in that cluster. Finally, a simple multi-label SVM classifier is used for actual prediction.

LEGINUS *et al.* (2012) present a different take on clustering because, unlike the previous examples, they apply clustering not on the examples but on the labels. In other words, they cluster labels into label clusters, based on similarity. They use clustering techniques such as k-Means, Spectral k-Means and Mean Shift and report gains in efficiency and accuracy when using these representations, when compared to using the full data.

[21] One may argue that this is similar to training paragraph vectors (as in LE & MIKOLOV (2014)), wherein neural networks are used to force the learning of vector representations that minimize the relationship between paragraph vectors and words therein.

[22] See JAIN (2010) for a comprehensive overview and summary on data clustering.

[23] This method is explained in detail in section 5.2.

[24] This is known as image segmentation (HARALICK & SHAPIRO, 1985).

Chapter 4

Proposal and Experiment Outline

4.1 Proposal

Initially, our proposal was to analyze multiple multi-label classification algorithms and verify how they perform when applied to the task of predicting social tags in broad folksonomies. We chose multi-label classification techniques because they are by far the most common method applied to social tag prediction (as evidenced in our literature review) and also because there is already a large body of work dedicated to this particular form of machine learning[1]

However, after reading many articles where this particular type of technique is applied to social tag prediction, we noticed that most use, in fact, a *label ranking* approach, which is very similar to multi-label classification, but where continuous, rather than real-valued, scores are assigned to each label.[2]

Since it quickly became apparent that label ranking was indeed the most widely used approach, we have chosen to slightly modify our proposal; we changed our focus from multi-label classification to multi-label ranking. This has enabled us to compare methods that are actually in use in the literature and see how these results fare in comparison to those reported by other authors.

Our final proposal is twofold:

Firstly, we would like to verify the performance of a group of social tag prediction methods. We will use techniques that are widely used

Additionally, we intend to test these on two different datasets, which differ on key metrics such as average number of tags per resource, total number of tags, total number of resources, etc. This will enable us to compare the performance of these methods under different settings. If we had used a single dataset to experiment with, we could risk choosing one that unfairly benefits one method to the detriment

[1] See TSOUMAKAS & KATAKIS (2007) for a comprehensive overview of the subject.

[2] More about the relationship between multi-label classification and label ranking on section 3.2.4

of others. In other words, this is a way to reduce any possible bias that could result from using a single dataset to compare these methods.

We consider this an important issue because the tag vocabulary and the folksonomy as a whole exhibits *emergent semantics* (CATTUTO *et al.*, 2007; KÖRNER *et al.*, 2010) due to its collaborative nature. This means that the characteristics of such systems may vary in multiple, sometimes unpredictable ways.

Among the characteristics datasets may differ in, we can count:

- **Total number of resources**: The total number of resources in a dataset may affect the outcome of many prediction approaches, particularly those that need many samples to learn from.

- **Total number of unique tags**: A dataset where resources are tagged using a limited tag vocabulary will probably be more amenable to tag prediction, independently of the approach used.

- **Average number of tags per resource**: We suspect that the number of tags each resource has been assigned will have an impact on classification and ranking. This is because it may be easier to return valid tags if there are more to choose from (for a given resource).

- **Minimum and maximum number of tags per resource**: The fact that some datasets allow some resources to have either zero or an unlimited number of tags may affect the performance of ranking approaches that rely on some sort of calculated *threshold* or cut-off value to define which tags are predicted.

- **Number of resources per tag**: The number of times each individual tag was assigned will probably be important because if there are too few examples some approaches may be unfeasible.

Secondly, we would like to verify to what extent the technique introduced by SHEN *et al.* (2009), namely *Multi-Instance Multi-label Learning for Automatic Tag Recommendation* works when applied to other kinds of textual features other than TF-IDF-weighted bag of words.

We propose this experiment because there are multiple techniques (mostly linear methods, such as Logistic Regression and SVM with a linear Kernel) that work well with bag of words due to their sparse nature (LI *et al.*, 2015; WEI HSU *et al.*, 2010), but may struggle with text representations where each document is represented not by a sparse feature vector but by a dense one instead.

We would therefore like to investigate if and in what way the results obtained using multi-instance learning for sparse vectors extrapolate for dense and otherwise different text representations.

One way to find that out is to try the aforementioned method with other representations for documents that have been used in the literature, which turn documents into *dense* feature vectors, as follows:

- **LDA Topic Probabilities**: As suggested in the original article that introduced LDA (BLEI *et al.*, 2003), one can use topic probabilities for each topic as a *representation* for a document. This is in spite of the fact that LDA is mainly a non-supervised technique to extract topic densities from a text corpus.
- **IDF-weighted Average of Word Embeddings**: Word embeddings[3] are fixed-dimension, dense representations for individual words. It has been recently suggested that one used the IDF-weighted average of word embeddings in a document as a representation for that document (JÚNIOR *et al.*, 2017; ZHAO *et al.*, 2015). Furthermore, this strategy has been established to work reasonably well according to many authors (ARORA *et al.*, 2017; WIETING *et al.*, 2015).

For the same reasons as in proposal 1, we will conduct these experiments on two datasets with different characteristics, to avoid biased results.

4.2 Datasets

For verifying our initial proposals, we envisioned a set of experiments on real world datasets. We decided to use at least two data sources with significant previous usage in the literature, so we could easily compare our results to previous experiments.

In addition, we wanted to see our proposed methods fares in tag prediction tasks in datasets with different characteristics. We took into account dataset metrics such as the average number of tags assigned to each resource, total number of resource, total number of unique tags, and so on.

4.2.1 Dataset 1: Delicious t-140

This dataset has been created during June 2008 for ZUBIAGA *et al.* (2009), for the task of Content-based Clustering.

4.2.1.1 Construction

This dataset was constructed using by subscribing to the 140 most popular tags on the Delicious.com website[4] between April 07, 2008 and April 12, 2008. Every time

[3] One of the seminal articles for word embeddings is BENGIO *et al.* (2003).

[4] Delicious was a popular online bookmarking website now inactive.

a URL was tagged using one of these 140 tags, the corresponding HTML document would be stored in the database (along with all tags assigned to it, even if they weren't in the top 140 set).

Once this dataset was collected, all tags having occurred in less than two documents were removed (so as to abide by the website's "common tag" definition and to remove overly noisy tags). Also, webpages written in languages other than English were also removed from the dataset.

After these initial steps, the dataset totalled 144,574 unique documents and 67,104 unique tags.

4.2.1.2 Preprocessing

Following literature conventions, we added a few pruning and preprocessing steps to this dataset, so as to make it more amenable to training models on.

We removed documents that had only been tagged with a single unique tag. We also removed from the dataset all documents which has been tagged by only a single user. More importantly, we removed from the dataset all tags which had been used in less than 10 separate documents.[5] These pruning steps brought the total number of documents down to 143,716 and the number of distinct tags to 9,184.

As for text preprocessing, we normalized all tags by applying lowercasing and removing special characters. As far as the textual contents of HTML pages are concerned, we removed HTML tags to arrive at a *clean* version of the dataset, again following literature convention,

Table 4.1: Dataset Statistics: Delicious t-140 (after pruning and preprocessing)

Total number of Resources	147,716
Total number of unique tags	9,184
Average number of tags per resource	13.12
Minimum number of tags per resource	1
Maximum number of tags per resource	25
Average number of resources per tag	205.24
Minimum number of resources per tag	10
Maximum number of resources per tag	26,603

[5]Such *tag-pruning* reflects standard practice in many works dealing with tag prediction, especially as related to broad folksonomies.

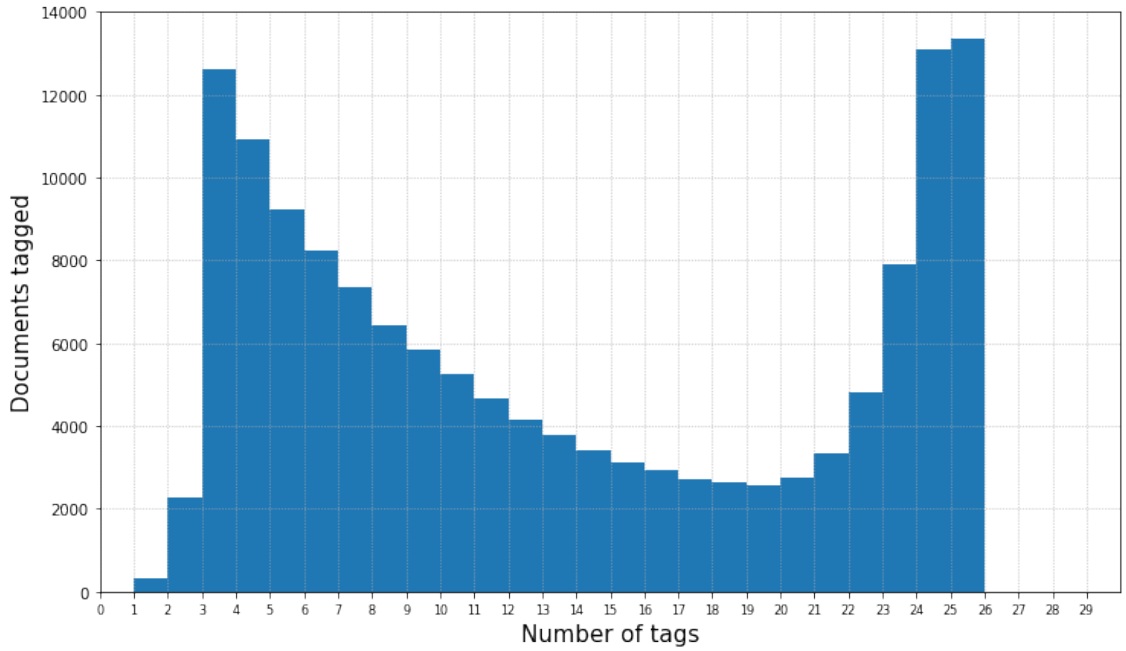


Figure 4.1: Distribution of the number of unique tags assigned to each document in the Delicious t-140 dataset (after pruning and preprocessing).

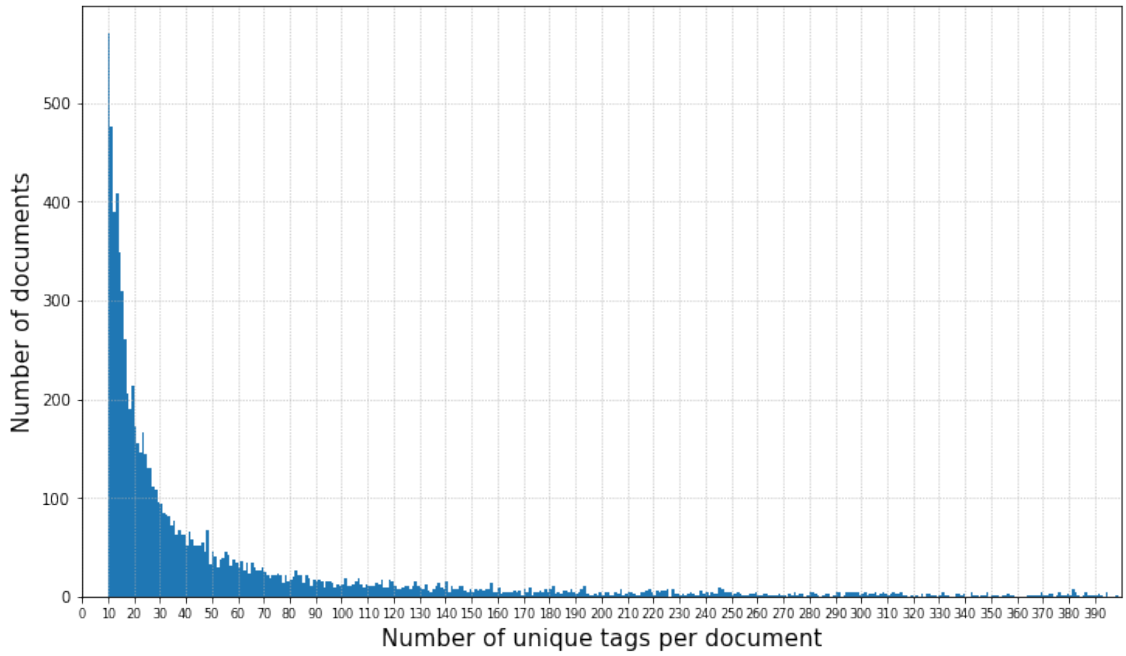


Figure 4.2: Distribution of the number of documents each tag was assigned to in the Delicious t-140 dataset (after pruning and preprocessing, not counting multiple assignments).

4.2.2 Dataset 2: Movielens 20M + IMDB Synopses

For our second dataset we chose one which, as previously explained, had different characteristics compared to the first dataset. We did this to verify whether (and to

what extent) our methods and other methods perform in datasets which differ with respect to metrics such as average number of tags per document, total number of tags, etc.

Once again, we wanted to choose data from sources which have been often used in the literature. With that in mind, we chose to work with a MovieLens dataset and with movie synopsis data from the International Movie Database.

While it is true that combining both datasets yields another dataset which different from the first two, there are examples in the literature (KATARIA, 2016; PERALTA, 2007) where these two datasets were combined.

Table 4.2: Dataset Statistics: MovieLens 20M + IMDB Synopses (after pruning and preprocessing)

Total number of Resources	6,710
Total number of unique tags	2,138
Average number of tags per resource	12.21
Minimum number of tags per resource	1
Maximum number of tags per resource	189
Average number of resources per tag	38.33
Minimum number of resources per tag	10
Maximum number of resources per tag	854

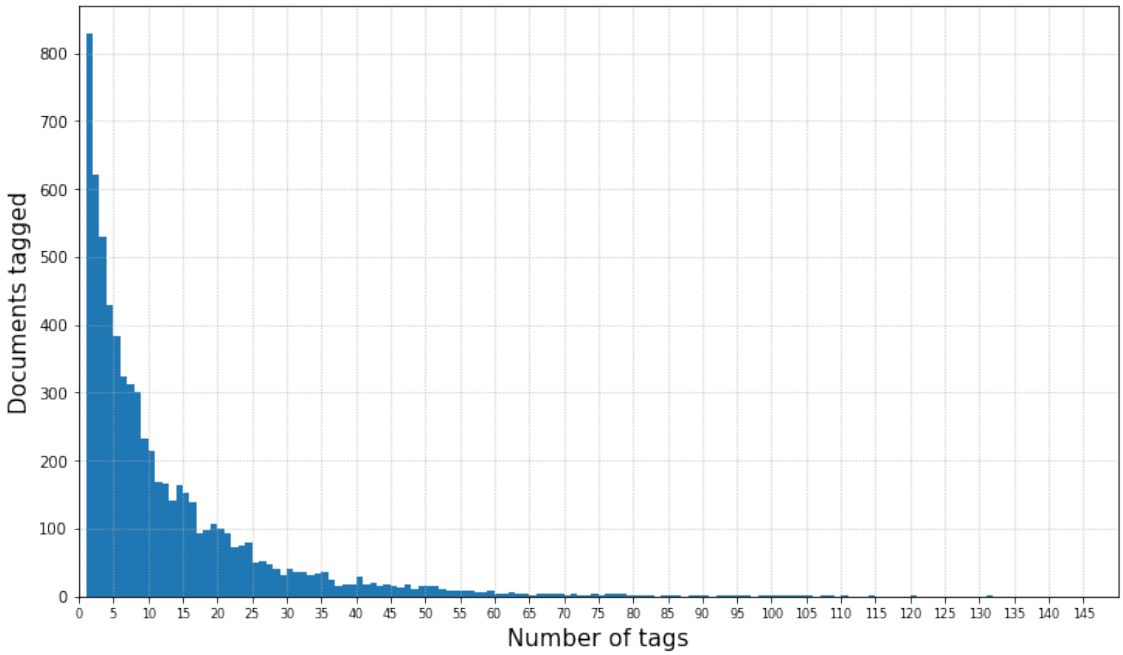


Figure 4.3: Distribution of the number of unique tags assigned to each document in the Movielens 20M + IMDB Synopses dataset (after pruning and preprocessing).

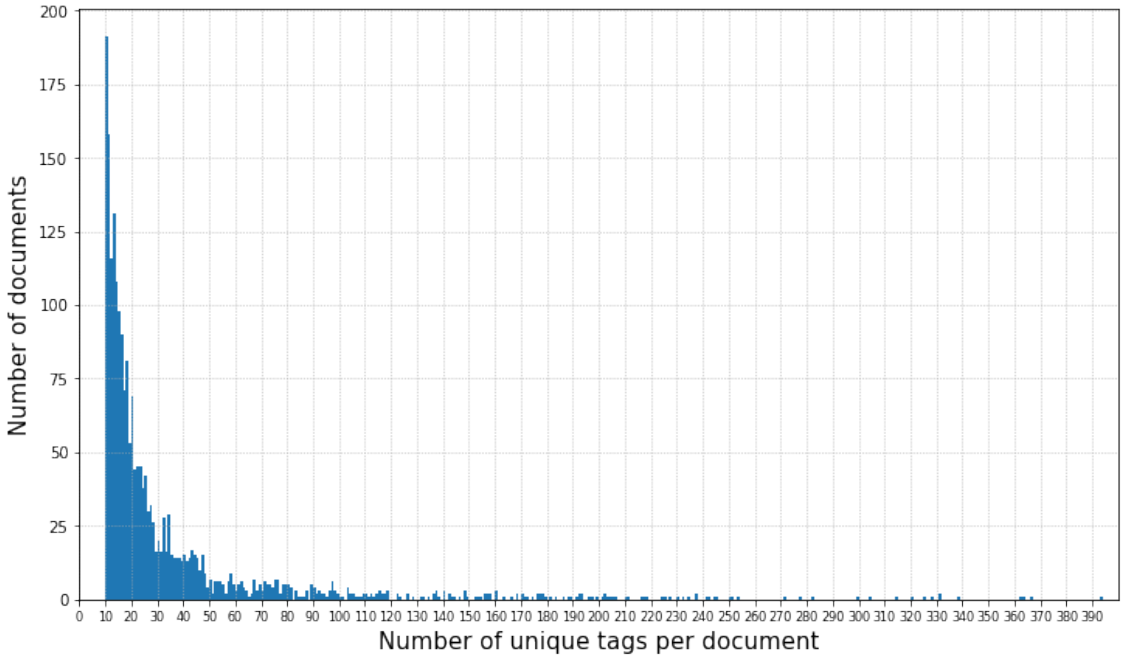


Figure 4.4: Distribution of the number of documents each tag was assigned to in the Movielens 20M + IMDB Synopses dataset (after pruning and preprocessing, not counting multiple assignments).

4.2.2.1 Construction and Preprocessing

The first part of the dataset was obtained at https://grouplens.org/datasets/movielens/20m/. This download package in-

cludes a file with every tag assignment until October 17, 2016, for movies in the MovieLens website.

We preprocessed this dataset by normalizing all tags: lowercasing and removing special characters. In addition, we removed from the dataset all tags that occurred in less than 10 documents, following literature convention.

The download package includes a file that matches each MovieLens movie ID with the corresponding movie ID on the Internet Movie Database (IMDb) website [6]. So, for each movie in the MovieLens dataset, its synopsis (when available) was manually scrapped from the IMDB website, using the *Scrapy* [7] tool.

After crawling the website for the matching movie synopses, we saved the results and filtered out movies with non-english synopses.

4.3 Experiment Outline

In the following subsections, we will briefly explain the underlying reasons for the way we have setup our experiments.

4.3.1 Project Structure

In the following subsection, we will briefly describe the actual software project created to fulfill the objectives described in earlier chapters.

4.3.1.1 Frameworks and Libraries used

We have chosen to use the Python programming language, due to its ease of use as well as the widespread availability of scientific libraries.

For exploratory data analysis and training all models, we have used tools such as *Numpy*, *Scipy*, *Pandas*, *Scikit-learn* and *Matplotlib*. All of these were used on top of *Jupyter* notebooks, to make all process easiily viewable and auditable.

For text preprocessing, we have used a parallel processing framework called *Apache Spark*, due to the size of the datasets. All workloads were executed on top of Amazon Web Services (AWS) infrastructure.

For crawling the IMDB website we have used a tool called *Scrapy*.

Finally, for training and using word embeddings we have used the *Gensim* framework for topic modelling.

A more detailed description of the project structure can be found in Appendix A.

[6] http://www.imdb.com/
[7] https://scrapy.org/

All code for the experiments is available under https://github.com/queirozfcom/auto-tagger/tree/master/social-tags.

4.3.2 Method Selection

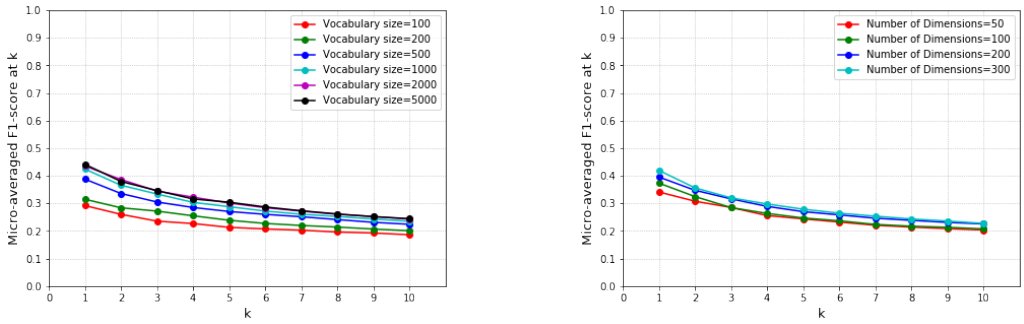
These experiments are meant to address proposals **1)** (performance comparison of multiple tag-predictions approaches in two very different datasets) and **2)** (comparing the MIMLSVM tag prediction method using sparse and dense features).

In order to have representative and non-biased experiments, we have chosen to use methods that were **a)** widely used in practice, **b)** different from other methods or **c)** both.

4.3.3 Hyperparameter Tuning

As is commonplace in most machine learning tasks, we have, for each experiment, tried a combination of hyperparameters for each method we have applied. We have used *grid search*, probably the most common way to conduct hyperparameter search, to search for good configurations for the problems at hand. The actual search was done on a sample (generally 30% of the full data) and the victor parameters used to train the model on the full datasets.

With respect to text-specific machine-learning, there is also the question of how to tune some feature extraction procedures. Once again, we have tried to emulate what has been done by other authors we have reviewed while also taking into account standard practice in the Natural Language Processing (NLP) field. We consider the two most important choices to be **a)** the number of words to use in BOW representation and **b)** the number of dimensions to use for word embeddings. We have conducted two simple comparisons to help us make appropriate choices for these parameters, taking into account both accuracy but also more practical matters such as training time and memory needed.



(a) Comparing performance using different vocabulary sizes (OvR SVM).

(b) Comparing performance using different embedding dimensions. (OvR SVM)

Figure 4.5: Comparing choice of hyperparameters for feature extraction. Using Dataset 2 for illustrative purposes.

Based on the above tests, we have concluded that using a vocabulary with only 500 as the number of words and 100 as the embedding dimension represents a good trade-off between performance and training time and complexity. In other words, we consider these to be enough to enable comparing methods while not incurring long training times and extreme memory consumption.

4.3.4 Metrics and Evaluation

Problems with multi-label data (the type we have in this work) can be approached in one of two ways (ILLIG *et al.*, 2011; TSOUMAKAS *et al.*, 2010): as **multi-label classification** or **label ranking**. The first type produces models that output a partition of labels (relevant/non-relevant) for each example. Conversely, label ranking implies training models that output an *ordering* of labels for each instance.

In this work, we have chosen to frame social tag prediction as a *label ranking* problem. This follows standard practice in the literature but we also deem it more useful for real world tasks such as displaying a (finite) number of tag suggestions/predictions to users in an STS. In other words, the output of our classifiers will be a list of tags ranked in decreased order or relevance.

A wide variety of evaluation metrics[8] is used in label ranking. Among the many articles reviewed for this dissertation, we cite the following as the most commonly-used metrics in this domain.

4.3.4.1 Average Precision and Mean Average Precision

Average Precision (AP) is a widely-used[9] metric to measure the result of a single list of ranked labels or a list of ranked documents (in an information retrieval setting).

[8]For extended commentary on ranking metrics see SOKOLOVA & LAPALME (2009) and KISHIDA (2005).

[9]See BUCKLEY & VOORHEES (2000) for a comprehensive study.

In general terms, AP measures, up to a cutoff value m, the precision achieved considering all labels up to label i:

$$AP_m = \frac{1}{m} \sum_{i=1}^{m} Precision@i \cdot \phi(i) \ , \tag{4.1}$$

where $Precision@i$ refers to the precision considering only the top i labels; $\phi(t)$ is an indicator function whose value equals 1 if predicted label at rank i is indeed a true label and 0 otherwise.

Now, when one wants to calculate AP over a whole dataset (as is out case), one can average AP over all documents for which we have predicted labels. This brings us to Mean Average Precision (MAP), which is calculated as follows:

$$MAP_m = \frac{1}{|D|} \sum_{d \in D} AP_m(d) \ , \tag{4.2}$$

where D is the set of documents for which labels are to be predicted.

4.3.4.2 Micro-Averaged F1 @k

Another very commonly-used metric, and the one that we have chosen to work with, is **micro-averaged F1-score @k**

This measure was chosen due to the problem we wish to consider (namely, label ranking) and the way we want to average the results over a given dataset. In addition, this metric is commonly used in articles we have reviewed.

The F1-score (a particular case of the more general *F-measure*, where β equals 1) is widely used in information retrieval problems related to search or ranking of results; it is the harmonic mean of precision and recall, given by:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.3}$$

which can be also written in terms of generic error metrics:

$$F_1 = \frac{2 \cdot true\ positive}{2 \cdot true\ positive + false\ negative + false\ positive} \tag{4.4}$$

With regards to *micro-averaging*, it refers to the way we report results for the whole dataset, be it training or validation.

When *macro-averaging* is used, equal weight is given to every class (label) in the dataset, which means that classes which occur only very rarely are given the same weight and very common classes when the full metrics over the dataset are calculated.

On the other hand, with *micro-averaging*, the individual metrics (true positive, true negative, false positive and false negative) are aggregated over the whole dataset, which is preferable in cases (such as ours) where the dataset is highly unbalanced.[10]

Finally, when metrics @k are considered, it simply means that only the results up to the k-th position are taken into account when gathering the results:

$$F_1 \ @k = \frac{2 \cdot true\ positive\ @k}{2 \cdot true\ positive\ @k + false\ negative\ @k + false\ positive\ @k} \tag{4.5}$$

This gives a more complete view of how the classifier works at different precision/recall levels, and can be easily visualized via graphical charts.

[10] I.e. some labels appear much more often than others.

Chapter 5

Experiments

5.1 Experiments for Proposal 1

In this section we present the results of experiments we conducted in order to empirically ascertain the difference in performance of several multi-label ranking methods, applied to social tag prediction, as detailed in Chapter 4.

For all experiments, we split the datasets into train/test sets in the proportion of 85/15. In other words, training and testing are done disjoint sets, so as to enable an unbiased estimate of the model's error rate.

5.1.1 TF-IDF weighted Bag-of-words Features, Binary Relevance + Linear SVM Classifier

In this experiment, we apply the commonly-used *Binary Relevance*[1] meta-estimator (TSOUMAKAS & KATAKIS, 2007) using a linear SVM classifier as underlying model.

This is a commonly-used technique for social tag prediction, as seen in CHEN *et al.* (2008); GOH *et al.* (2008); ILLIG *et al.* (2011); TAO & YAO (2016) among others.

[1] As previously noted, this method is also called *One-vs-Rest* because one classifier is trained for each separate category, or label.

5.1.1.1 Results on Dataset 1

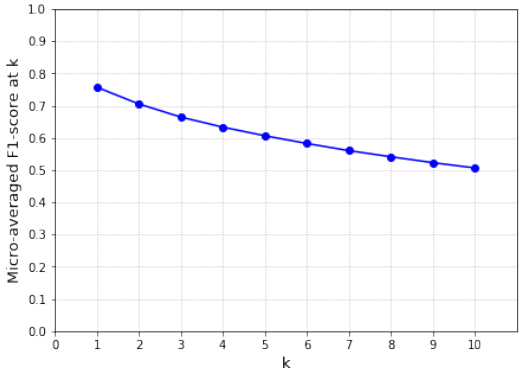


Figure 5.1: Results of applying Binary Relevance + Linear SVM with TF-IDF features on the Delicious t-140 Dataset (validation set scores shown)

5.1.1.2 Results on Dataset 2

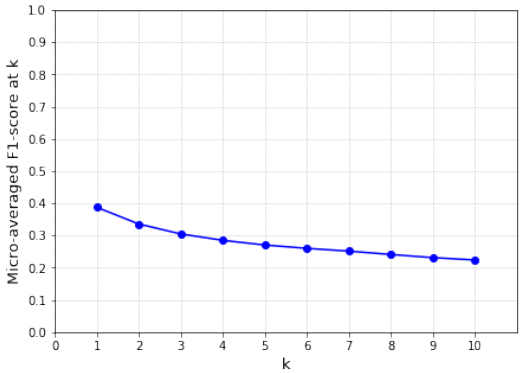


Figure 5.2: Results of applying Binary Relevance + Linear SVM with TF-IDF features on the Movielens Dataset (validation set scores shown)

5.1.1.3 Discussion

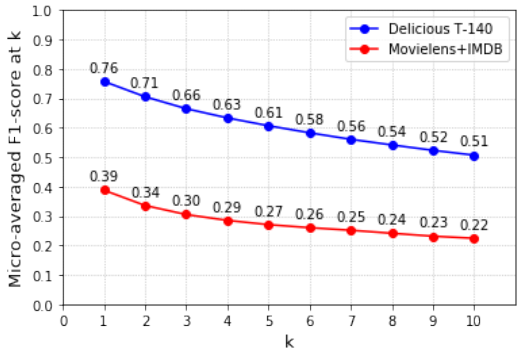


Figure 5.3: Binary Relevance, Linear SVM with TF-IDF features: Compared results (validation set scores)

As expected, the results in Dataset 1 were far better than those for Dataset 2, due to the differences in the tag distribution for both datasets.

It is interesting to note that the decrease in scores for Dataset 1 is somewhat more pronounced than in Dataset 2.

5.1.2 TF-IDF weighted Bag-of-words Features, k-Nearest Neighbours Classifier

The k-Nearest Neighbors is a very popular machine learning method that can be used both for classification and for regression. It consists in simply calculating the distances (assuming an n-dimensional representations) to every other instance, at *inference time*[2]. Then, each neighbor up to k is treated as a source of information to help predict the class for the query instance.

With respect to tag prediction, multiple (CHARTE *et al.* (2015); CHIDLOVSKII (2012); MARTÍNEZ *et al.* (2009); ZHANG *et al.* (2015)[3] to cite but a few) authors have applied some form of neighbor-based classifier to predicting tags for a query resource.

In general, they proceed by finding nearest neighbors based on the resource's vector representation, as per the usual algorithm. Then, each neighbor's binary tag vector is added up and tags which are more commonly seen in the query instance's neighborhood are suggested.

Since we only want to use this method as a baseline, we implemented the most basic version thereof, namely simple, unweighted k-NN. Furthermore, we ran grid search over the method's hyperparameters, namely k, the number of neighbors to consider and also over the distance metric to use (cosine, euclidean, manhattan, etc).

[2]Methods such as k-NN are called *lazy* methods because they need no training, as they defer all processing until actual inference is made.

[3]CHARTE *et al.* (2015) have applied an approach very similar to ours, namely using multi-label k-NN to classify text into multiple tags, using TF-IDF representation.

5.1.2.1 Results on dataset 1

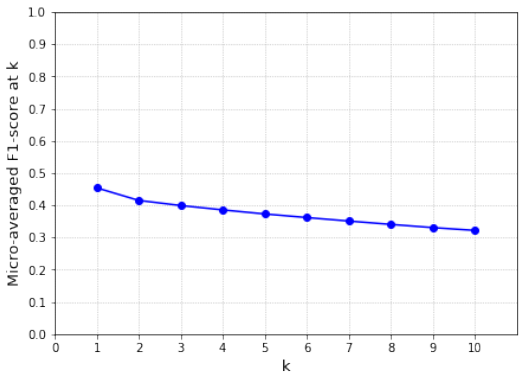


Figure 5.4: Applying k-NN on the Delicious Dataset, using TF-IDF weighted bag-of-words representation (validation set scores shown)

5.1.2.2 Results on Dataset 2

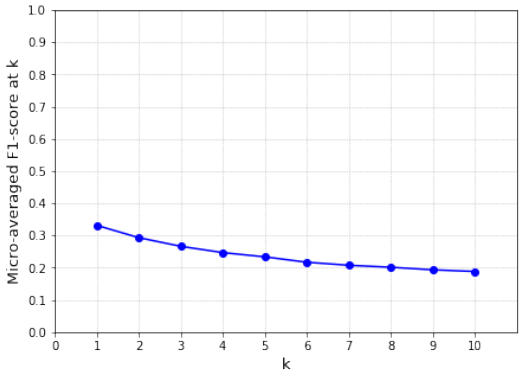


Figure 5.5: Applying k-NN on the Movielens Dataset, using TF-IDF weighted bag-of-words representation (validation set scores shown)

5.1.2.3 Discussion

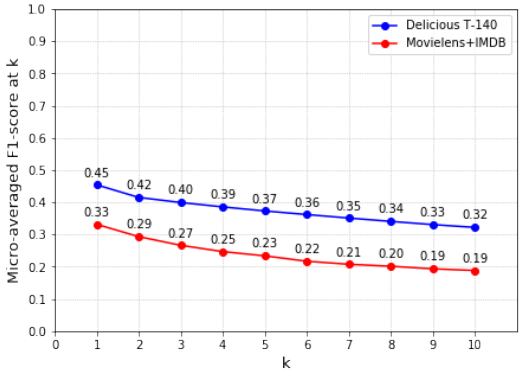


Figure 5.6: k-NN with TF-IDF features: Compared results (validation set scores)

Although the results were satisfactory, one can see that the difference in performance on both datasets is not as large as in the previous example. One reason for that may be that the large number of neighbours (found by model selection via grid search) may act as a regularizer, decreasing the variance on out-of-sample examples but at the cost of a higher bias.

We would like to note that, surprisingly, using a *weighted* variant did not increase performance on this task. In other words, weighing the contribution by the inverse of the distance to each neighbor did not increase the accuracy of the model.

5.1.3 TF-IDF weighted Bag-of-words Features, Topic Distances

In this approach, which has been suggested by CHOUBEY (2011),we first train a topic model on train set documents using Latent Dirichlet Allocation (LDA) (BLEI *et al.* (2003)). Then, at query time, we calculate the topic distribution for the query document and also the single most similar train set document, as measured by the Kullback-Leibler Divergence (KL-Divergence, KULLBACK & LEIBLER (1951)) between the topic distributions of the documents. Finally, the tags used in the found document are used as suggestions for the unlabelled query document.

5.1.3.1 Results on dataset 1

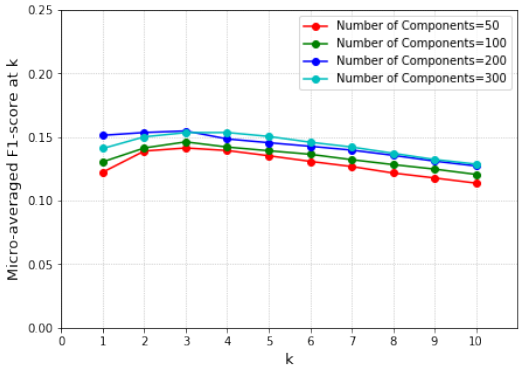


Figure 5.7: Applying Topic Distances on the Delicious Dataset, with varying values for the choice of LDA components

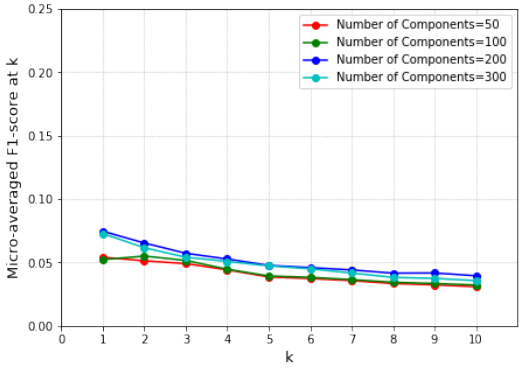


Figure 5.8: Applying Topic Distances on the Movielens Dataset, with varying values for the choice of LDA components

5.1.3.2 Discussion

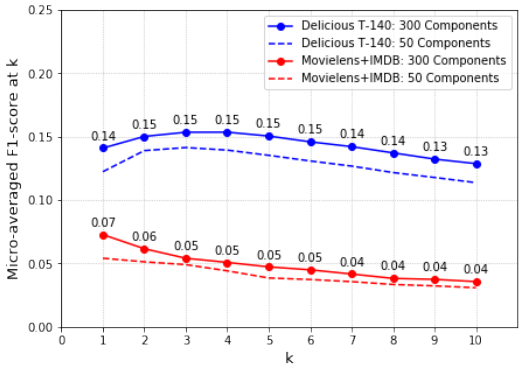


Figure 5.9: Topic Distances: Compared results (validation set scores). Best and worst results for each Dataset shown for comparison.

While the results were overall worse than previous classifiers, the overall pattern of dataset 1 (Delicious) performing better than dataset 2 was maintained. Notably, however, the difference is now much more pronounced (in relative terms), standing at up to 300%.

It is worth mentioning that the results achieved are close to what the original authors', lending credibility to the fact that this method performs very poorly overall, not just on specific datasets and/or specific conditions.

5.1.4 TF-IDF weighted Bag-of-words Features, Topic Words

In this approach, also suggested by CHOUBEY (2011), one trains an LDA topic model on documents in the train set. At test time, the topic distribution for each query document is calculated with the trained model. Then, the most representa-

tive words[4] for the most representative topic are suggested as tags for the query document.

5.1.4.1 Results on dataset 1

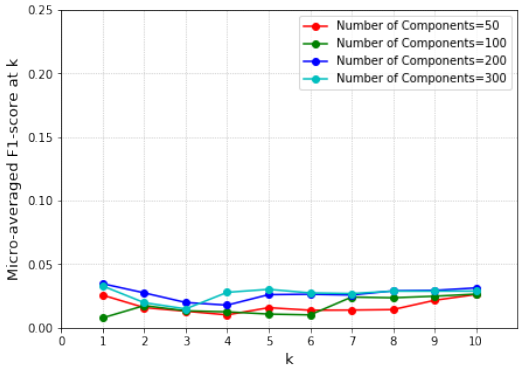


Figure 5.10: Applying Topic Words on the Delicious Dataset, with varying values for the choice of LDA components (validation set scores shown)

5.1.4.2 Results on Dataset 2

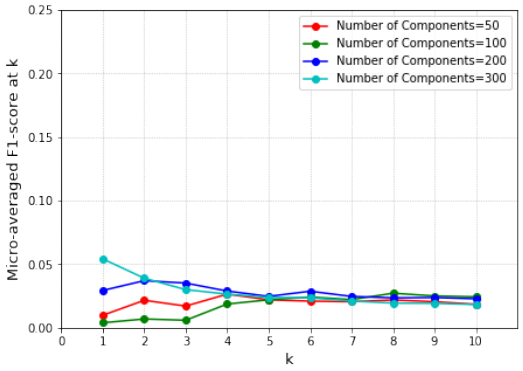


Figure 5.11: Applying Topic Words on the Movielens Dataset, with varying values for the choice of LDA components (validation set scores shown)

[4]Only words that are in the actual tag vocabulary are used.

5.1.4.3 Discussion

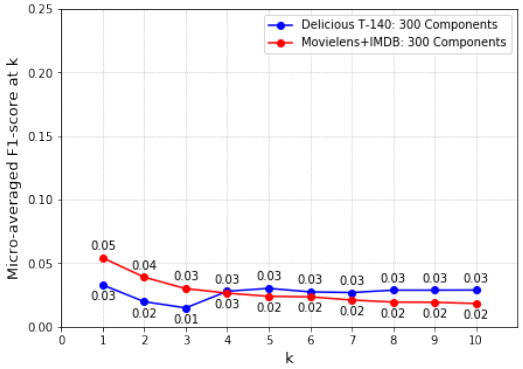


Figure 5.12: Topic Words: Compared results (validation set scores) using the best choice for the number of components.

Once again, the results are not very good (in comparison to classifiers such as SVM, used previously). These, however, resemble results in the original source (CHOUBEY, 2011). Notably, there doesn't seem to be any consistent difference when one dataset is compared to another, or as k grows. This may indicate that this method is not effectively learning much.

5.1.5 LDA Topic Probabilities, k-nearest Neighbours Classifier

Although Latent Dirichlet Allocation (LDA) (BLEI *et al.*, 2003) was originally created as a means to infer representative words for topics in corpora, it can be (and frequently is) used to extract features for documents. In fact, this approach was used and suggested in the original paper itself.

In other words, LDA can be used as a form of dimensionality reduction to reduce the size of feature vectors[5] from V to k, respectively the vocabulary size and the number of components in the LDA model.

Using these topic probabilities as features, we can then proceed onto classifying the documents using any classifier we wish. We have chose to use two classifier for this task: **a)** a simple k-nearest Neighbours Classifier so as to enable comparison between using LDA features and using bag-of-words features and **b)** (in the next subsection) an SVM classifier, as suggested in the original LDA article.

[5]Assuming an original bag-of-words representation without trimming the number of words used.

5.1.5.1 Results on dataset 1

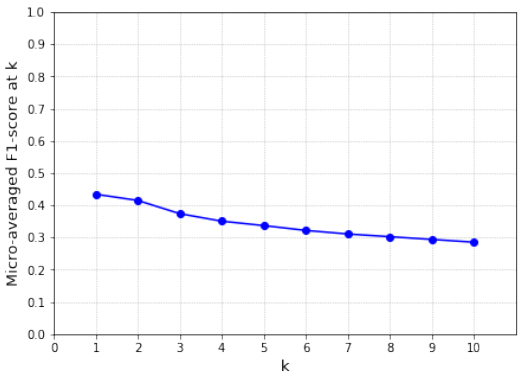


Figure 5.13: k-Nearest Neighbor classifier on the Delicious dataset, using LDA topic probabilities as features (validation set scores shown).

5.1.5.2 Results on Dataset 2

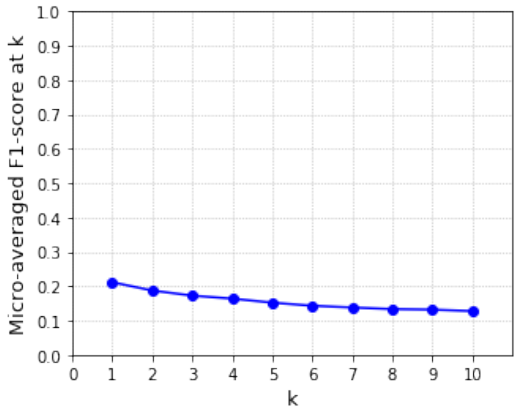


Figure 5.14: k-Nearest Neighbor classifier on the Movielens dataset, using LDA topic probabilities as features (validation set scores shown).

5.1.5.3 Discussion

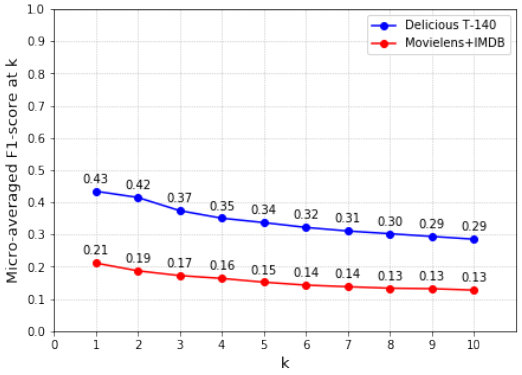


Figure 5.15: k-NN using LDA features: Compared results (validation set scores).

It is interesting to note that, although only 50 components are used in this setup, we get 0.43 as F-1 score, compared with 0.45 when using k-NN with 500-dimensional bag-of-words features. In other words, we were able to reduce the dimensionality[6] of the problem while losing just a bit of performance.

Interestingly, however, the compared results for the second dataset, namely, Movielens+IMDB, was markedly worse; 0.21 using LDA features vs 0.33 using bag-of-words features.

5.1.6 LDA Topic Probabilities, SVM classifier

As mentioned on the previous subsection, we will compare results between both datasets using LDA as a simple dimensionality reduction step on top of TF-IDF weighted bag-of-words features. We will use an SVM classifier, as suggested in the original LDA paper by BLEI *et al.* (2003).

However, since the features are now of a *denser* nature, we will add other types of kernels to the hyperparameter search space, namely Radial Basis Function (RBF) and a also a polynomial kernel, in addition to the default linear kernel.

5.1.6.1 Results on dataset 1

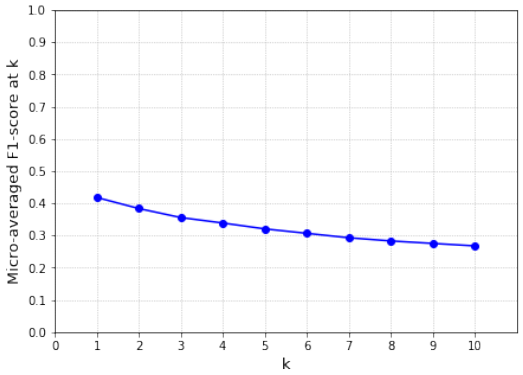


Figure 5.16: SVM classifier on the Delicious dataset, using LDA topic probabilities as features (validation set scores shown).

[6]Therefore also reducing the training and testing time, processing power needed, not to mention better generalization due to using a less complex model.

5.1.6.2 Results on Dataset 2

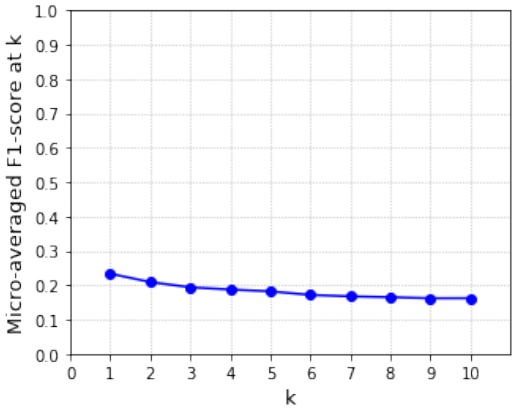


Figure 5.17: SVM classifier on the Movielens dataset, using LDA topic probabilities as features (validation set scores shown).

5.1.6.3 Discussion

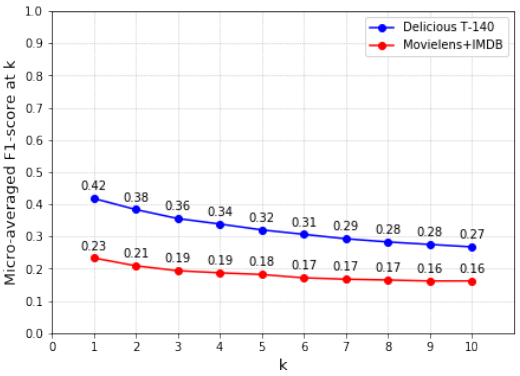
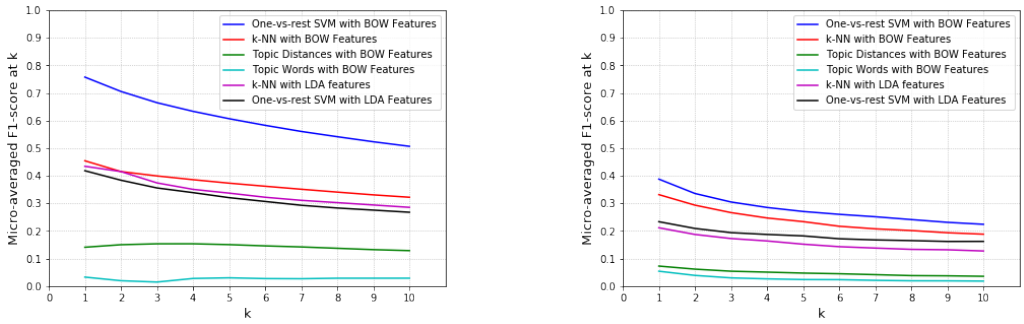


Figure 5.18: SVM using LDA features: Compared results (validation set scores).

Once again, we see that the difference in outcomes is large, of up to 80%. This is surprising in light of the fact that the best results for each dataset (as obtained via grid search) used just 5 LDA components.

In other words, we were able to provide good performance for this experiment setup using a mere 5 dimensional in place of the original 500 dimensions of the bag-of-words vectors. Note that an RBF kernel was used in this section.

5.1.7 Final Results and discussion



(a) Full comparison of all techniques used on dataset 1: Delicious T-140.

(b) Full comparison of all techniques used on dataset 2: Movielens+IMDB.

Figure 5.19: Full comparison of all techniques used for proposal 1 (validation set scores).

As we had initially expected, results in dataset 1 far outperform those in dataset 2, Delicious T-140 and Movielens+IMDB respectively.

Table 5.1: Compared dataset statistics (after pruning and preprocessing)

	Dataset 1	**Dataset 2**
Total number of resources	147,716	6,710
Total number of unique tags	9,184	2,138
Average number of unique tags per resource	13.12	12.21
Minimum number of unique tags per resource	1	1
Maximum number of unique tags per resource	25	189
Average number of resources per tag	205.24	38.33
Minimum number of resources per tag	10	10
Maximum number of resources per tag	26,603	854

The experiments seem to confirm the intuitive explanation that dataset characteristics (as seen on the table above) do indeed affect the performance of classifiers, at least when measured with our metric of choice (micro-F1 @k).

Intuitively, we could claim that the results were better in dataset 1 because number of resources is much larger (there are many more examples to learn from),

the tag vocabulary is smaller (there are less tags to choose from at each prediction) and the number of tags assigned to each resource is capped at 25.

Another possible explanation is that documents in dataset 1 represent the resource itself (a webpage) while in dataset 2 the documents are but a *description* of the resource, not the resource itself (resources are movies). In a way, dataset 2 contains *secondhand information* about the resource.

In addition to the points above, dataset 2 is a mixture of two different sources[7], namely Movielens for the tags and IMDB for the movie plot summary. This may have had an additional effect on lowering the performance of the classifiers since there is a potential mismatch between the two sources.

5.2 Experiments for Proposal 2

Multi-instance Learning[8] is a technique (the name was first coined by DIETTERICH *et al.* (1997)), whereby a an instance in a traditional supervised learning problem is split into multiple so-called *bags*.

In other words, each individual sample in a dataset is represented not by a single feature vector but by a set thereof. For example, images may be represented as a bag of patches (ANDREWS *et al.*, 2003; MARON & RATAN, 1998), pharmacological drug molecules may be represented as a bag of configurations (ANDREWS *et al.*, 2003; DIETTERICH *et al.*, 1997).

(a) Traditional supervised learning (b) Multi-instance learning

Figure 5.20: Multi-instance learning works by representing a single example as multiple instances.

In 2006, ZHANG & ZHOU have adapted the multi-instance learning paradigm into the multi-label setting, in the context of *scene classification*. The main insight put forward by this work is that a multi-instance, multi-label (MIML) problem can be transformed into either **a)** a single-instance, multi-label task or **b)** a multi-instance, single-label task:

[7] See subsection 4.2.2 for a detailed explanation.
[8] Also called *Multiple-instance* Learning

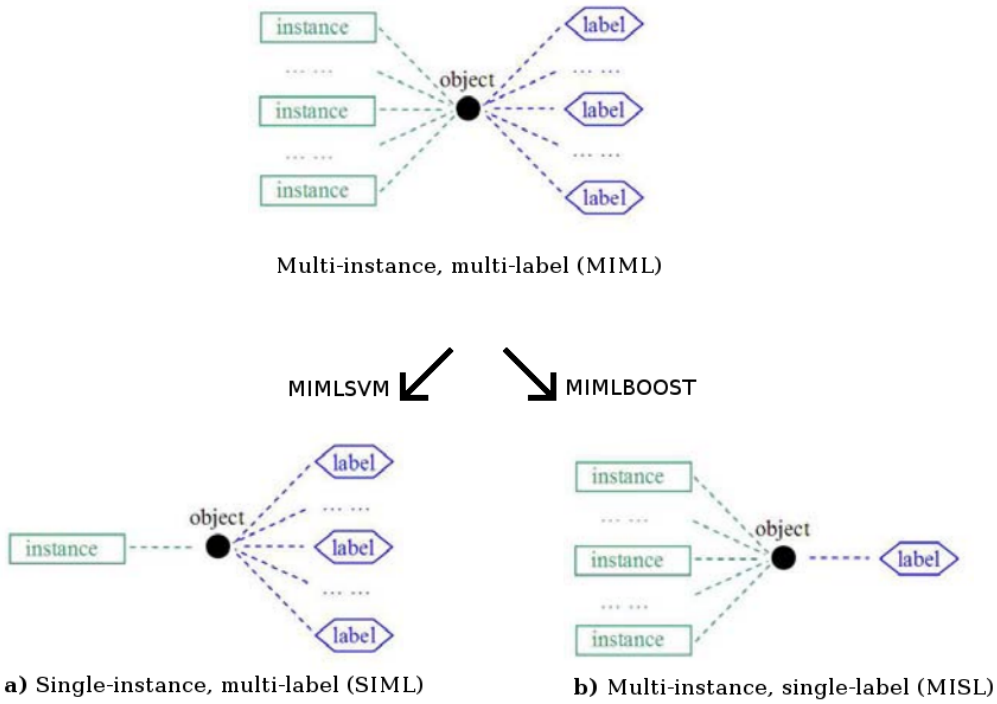


Figure 5.21: Original algorithm, devised by ZHANG & ZHOU (2006), transforms an MIML problem into either a SIML or an MISL problem, using MIMLSVM and MIMLBOOST techniques, respectively.

In 2009, SHEN *et al.* have applied multi-instance, multi-label (MIML) learning to the tag prediction problem. In particular, they have adapted the MIMLSVM algorithm from ZHANG & ZHOU (2006) to multi-label text classification.

The main idea here is that a single textual document may be split into multiple *segments* via some kind of text segmentation algorithm. This makes it possible to view this problem as a multi-instance, multi-label (MIML) problem, where each segment represents one of many instances for a single document.

Each document is split into segments using a well-known text segmentation algorithm called *TextTiling* (HEARST (1994)). Then, these segments are vectorized into bag-of-words vectors. In order to turn the multiple segments into a single instance, the authors use k-medoids clustering based on the Hausdorff distance (EDGAR (2008)). Finally, an SVM classifier[9] is applied on to the transformed dataset.

[9]Configured so that it predicts a real-valued score for each tag instead of a binary prediction.

Algorithm 1: MIMLSVM applied to Tag Prediction (SHEN *et al.*, 2009)

input : A set D of text documents

output: A trained SVM model to rank tags y_d for every d in D

Part I: Building a Single-Instance Dataset

foreach *document* d *in* D **do**

```
// split document into segments
segments_d ← TextTiling(d)

// transform each segment into a vector of features
vectorizedSegments_d ← extractFeatures(segments_d)

// apply k−medoids clustering algorithm to the segments of d.
// note that features_d is now a single-instance vector
// because Hausdorff Distance was used in clustering
features_d ← kMedoids(vectorizedSegments_d)

// this becomes a single row in the new D' dataset
D'_d ← features_d
```

Part II: Training an SVM Classifier on D'

Train a *Ranked* SVM algorithm on the transformed features in D'

The objective of the experiments in this section is to ascertain whether (if at all) the original results generalize to other kinds of features.

As before, we split the datasets into train/test sets in the proportion of 85/15. In other words, training and testing are done disjoint sets, so as to enable an unbiased estimate of the model's error rate.

5.2.1 MIMLSVM with IDF weighted Bag-of-words features

The following is the original version of the MIMLSVM algorithm, i.e. using TF-IDF weighted Bag-of-words features. Values for all hyperparameters were found via grid search on a sample of the full dataset.

5.2.1.1 Results on Dataset 1

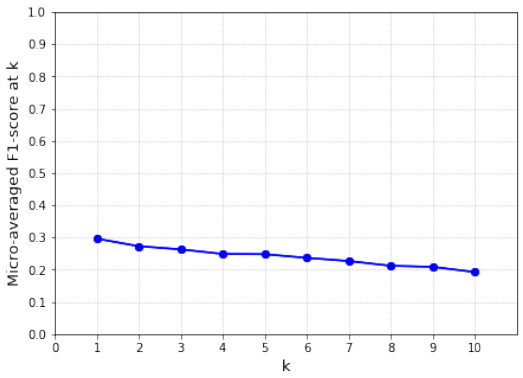


Figure 5.22: MIMLSVM classifier applied on the Delicious dataset, using TF-IDF weighted bag-of-words features (validation set scores shown).

5.2.1.2 Results on Dataset 2

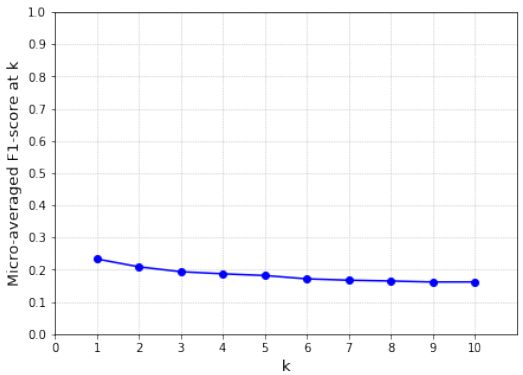


Figure 5.23: MIMLSVM classifier applied on the Movielens dataset, using TF-IDF weighted bag-of-words features (validation set scores shown).

5.2.1.3 Discussion

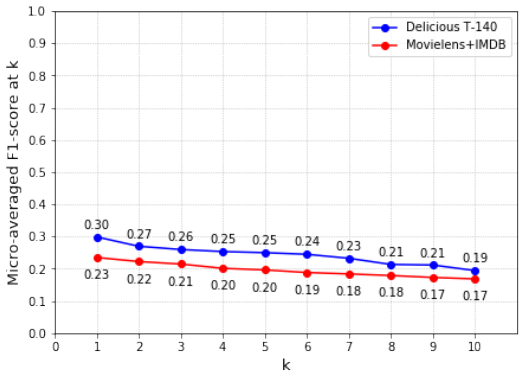


Figure 5.24: MIMLSVM with TF-IDF features: Compared results (validation set scores)

Although results for Dataset 1 continue to be better than dataset 2, we notice an interesting development: the difference in scores appears to be smaller than for previous classifiers (i.e. in part 1). We suspect this may be due to the fact that *TextTiling* works by identifying *topics* to split the documents by.

Dataset 1 is fully composed of actual sentences and phrases, while dataset 2 is made up of HTML source code (albeit with things like tags and javscript code removed). This may have caused the MIMLSVM technique to be better able to extract segment information from the former and not the latter.

Other than that, the results are comparable to those obtained by the original authors.

5.2.2 MIMLSVM with LDA Topic Probabilities as Features

Latent Dirichlet Allocation (LDA) BLEI *et al.* (2003) was originally thought of as an unsupervised method to learn the best way to infer latent topics for documents, based upon the distribution of words in them.

As mentioned before in section 5.1.5, the original LDA article itself suggests that topic probabilities be used as features to represent a document. This way, LDA can be thought of as a form of dimensionality reduction for documents, reducing the size of feature vectors from V, where V is the size of the vocabulary to D, where D is the number of components chosen when training the LDA model.

Each document is therefore represented as a feature vector of size D, where each $d_i \in D$ represents how much topic i is present is a document, as per BLEI *et al.* (2003). Since this vector is a dense vector, it serves out purpose of experimenting on using MIMLSVM on dense document representations.

5.2.2.1 Results on Dataset 1

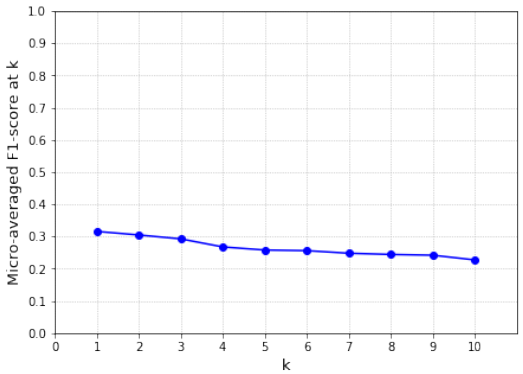


Figure 5.25: MIMLSVM classifier applied on the Delicious dataset, using LDA topic probabilities as features. (validation set scores shown)

5.2.2.2 Results on Dataset 2

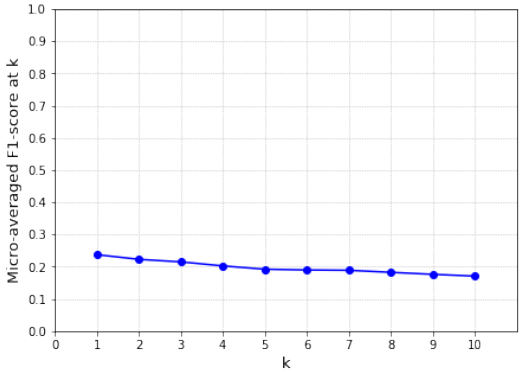


Figure 5.26: MIMLSVM classifier applied on the Movielens dataset, using LDA topic probabilities as features features.

5.2.2.3 Discussion

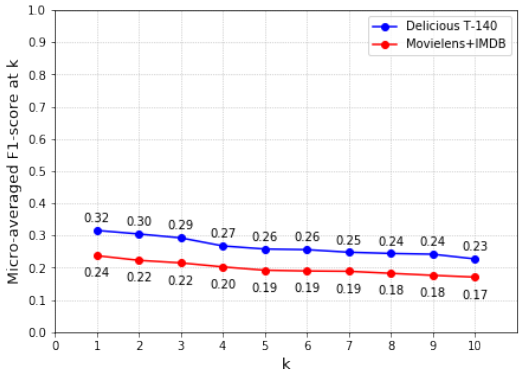


Figure 5.27: MIMLSVM with LDA features: Compared results (validation set scores)

This is the first experiment where we test out our original idea as detailed in proposal 2, namely whether MIMLSVM can generalize with non-sparse, i.e. dense, features.

The results seem to be only very slightly superior to those in the previous experiment using regular bag-of-words features. Initially, it doesn't seem to be the case that using more informative features, with less dimensions makes the prediction task much easier.[10]

[10]Note that hyperparameters and other choices such as SVM kernels and distance functions were kept the same so as to enable a fair comparison.

5.2.3 MIMLSVM with IDF-weighted Bag-of-embeddings Features

5.2.3.1 Results on dataset 1

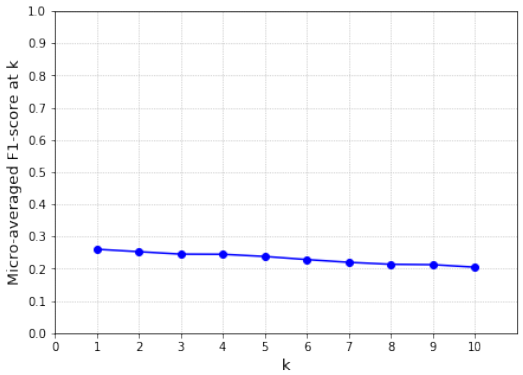


Figure 5.28: MIMLSVM classifier applied on the Delicious dataset, using IDF weighted bag-of-embeddings features (validation set scores shown).

5.2.3.2 Results on Dataset 2

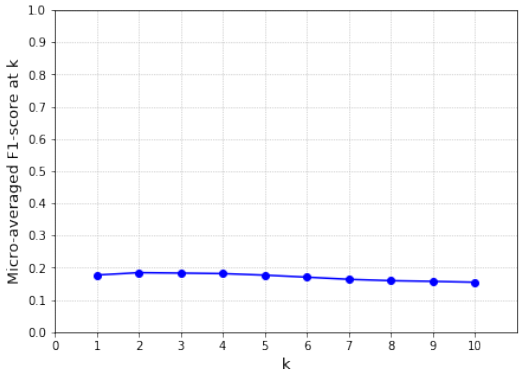


Figure 5.29: MIMLSVM classifier applied on the Movielens dataset, using IDF weighted bag-of-embeddings features (validation set scores shown).

5.2.3.3 Discussion

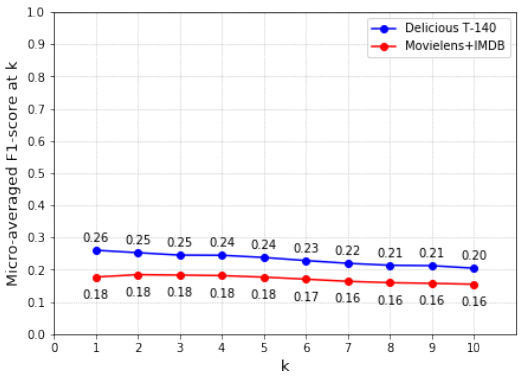
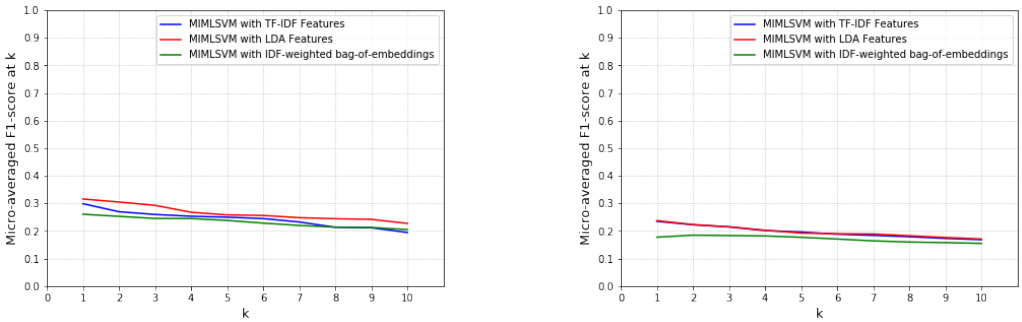


Figure 5.30: MIMLSVM with IDF weighted bag-of-embedding features: Compared results (validation set scores)

In this case, the switch to IDF-weighted bag-of-embeddings decreased classifier accuracy by a significant amount.

5.2.4 Final Results and discussion



(a) Full comparison of all MIMLSVM variants used on dataset 1: Delicious T-140.

(b) Full comparison of all MIMLSVM variants used on dataset 2: Moivelens+IMDB.

Figure 5.31: Full comparison of all techniques used for proposal 2 (validation set scores).

After conducting these experiments, it does look like the MIMLSVM algorithm can indeed be used for dense text representations, in both *well-structured* text with phrases and sentences as in dataset 1 and in looser, more unstructured text as in dataset 2.

Apparently the variant using LDA topic probabilities as features had a small but noticeable advantage over other cases for dataset 1, but not for dataset 2.

On the other hand, the variant using IDF-weighted bag-of-embeddings performed clearly worse in both cases.

Chapter 6

Conclusion and Future Work

In this chapter, we conclude the findings from the experiments and link them back to our original proposals and problem scope.

6.1 Conclusion

Overall, we consider the experiments to have been very informative in helping us answer the questions detailed in the problem scope, and planned in the proposals. Next, we give more detailed insights on both proposals.

6.1.1 Proposal 1

As seen in subsection 5.1.7, we verified that all methods perform consistently better on Dataset 1 as compared with Dataset 2. This may indicate that the first dataset, namely Delicious T-140 is *inherently* easier to predict tags for. Intuitively, this is probably related to the dataset characteristics outlined on tables 5.1 and 4.2.

Other factors may have played a role too: as mentioned before, dataset 1 is a *firsthand* dataset, in which we deal with the resources themselves, namely HTML source code for web pages. Dataset 2, on the other hand, contains textual *descriptions* of movies, not movies themselves (in which case we would need visual and/or audio features instead). When someone describes data such as video using text, some information will invariably be lost *in translation*.

In addition, we would also like to draw attention to the fact that the simplest algorithm, namely Binary Relevance with TF-IDF features and Linear SVM classifier yielded the best results for both datasets. This reminds us that, in the absence of more specific, semantic information about the problem domain, simple solutions which carry little to no assumptions about the data may be the safest approaches.

6.1.2 Proposal 2

Once more, as we have already briefly explained, the MIMLSVM algorithm (SHEN *et al.*, 2009) doesn indeed seem to generalize for other, non-sparse text represention; notably, using LDA topic probabilities as features (while keeping all other hypeparameters constant) seemed to yield a slight increase in prediction performance, at least for Dataset 1. The algorithm does not seem to fare as well with IDF-weighted bag-of-embeddings features, however, which has cause decreased performance across both datasets.

As in experiments for Proposal 1, the nature of the text in both datasets may have also played a role here; the segmentation procedure, namely *TextTiling*, is particularly sensitive to punctuation and other markers of prose text; applying this on HTML text (albeit *cleaned* HTML), may be stretching some assumptions this procedure was built for.

6.2 Threats to Internal and External Validity

The experimental setup detailed in this work may contain errors and inaccuracies inherent to any empirical undertaking. These can affect the results and jeopardize our conclusions.

By *threats to internal validity*, we mean issues that may compromise our confidence in saying the trust the results obtained. *Threats to external validity* are factors that may cause our approaches to fail to generalize well to other scenarios.

In the following list we state factors that may present threats to our experiment's internal or external validity.

- *Confounding variables*: There may be other, unaccounted for, variables that may affect the results obtained.
- *Generalization to other approaches*: It is possible that the approaches we selected for proposal 1 are not representative enough of all possible classifiers. We may have reached different results had we extended our experiments to even more methods.
- *Generalization to other STSs*: It may be that the two STS we selected, namely Delicious and Movielens+IMDB, display very specific features that have somehow biased the reuslts obtained.
- *Temporal effects*: It is possible that the results obtained here are only so because of the time frames involved; maybe if we had conducted the same experiments at some point in the future, the results would have been different.

6.3 Future Work

Although we were able to verify some aspects of the problems addressed, there remain many other areas which may be worthy of research.

6.3.1 Alternative similarity metrics for clustering multi-instances

The suggested approach uses the Hausdorff distance to calculate similarity between bags of instances, after a document has been split into segments. However, as suggested in the original article about scene classification (ZHANG & ZHOU, 2006), Hausdorff distance is but one possible mapping to convert multiple bags into a single feature vector prior to performing clustering.

Other distance metrics are available for comparing bags of vectors; HUTTENLOCHER *et al.* (1993) alone cite more than twenty variations that can be used under different conditions. Different metrics may yield different results, particularly when one considers not only sparse but also dense text representations.

6.3.2 Alternative clustering algorithms

While the k-medoids algorithm was used in the proposed approach, it remains to be seen whether other similar clustering algorithms could yield better results than those shown. In particular, similar, *centroid-based* clustering algorithms include k-means clustering (MACQUEEN, 1967), k-medians clustering (JAIN & DUBES, 1988) and k-means++ (ARTHUR & VASSILVITSKII, 2007).

6.3.3 Other classifiers for MIMLSVM

The choice of SVM for the classifier part of MIMLSVM seems to be reminiscent from the original paper by ZHANG & ZHOU (2006). The adaptation to text data introduced by SHEN *et al.* (2009) followed the example of the original implementation, but no reason was given for using SVM over any other classifier.

In particular for different types of features such as embedded representations, neural networks would be a natural choice, which could enhance results and make predictions more accurate.

6.3.4 Adapting algorithms from Computer Vision to Natural Language Processing

In Proposal 2, we applied a multi-label classification technique that had been originally designed for use with images. This indicates that there may be other ways to

adapt approaches from the area of Computer Vision to Natural Language Processing (NLP) problems.

This may be in part caused by similarities between the nature of images and text, among which:

- **Complex compositionality**: Compositionality for image parts and text parts is not trivial. In the same way that a picture of people and furniture may imply a higher concept (*i.e.* a home), composing words and/or phrases also displays a high level of abstraction. For instance, the phrase *New York* has hardly any connection to its composing parts (*i.e.* the individual words *New* and *York*).
- **Rich representation possibilities**: Computer Vision benefits from sophisticated ways of building representations for individual examples. For example, Convolutional Neural Networks (CNNs) (in particular multi-layer CNNs) build increasingly more complex representations for image data. (RAWAT & WANG, 2017)

  In the case of NLP, one can cite word and document *embeddings* as successful examples of the use of higher-level representations in machine learning. (MIKOLOV *et al.*, 2013)

Bibliography

AGRAWAL, R., IMIELIŃSKI, T., SWAMI, A., 1993, "Mining Association Rules Between Sets of Items in Large Databases". In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pp. 207–216, New York, NY, USA. ACM. ISBN: 0-89791-592-5. doi: 10.1145/170035.170072. http://doi.acm.org/10.1145/170035.170072 .

AIELLO, L. M., BARRAT, A., SCHIFANELLA, R., etal., 2012, "Friendship Prediction and Homophily in Social Media", *ACM Trans. Web*, v. 6, n. 2 (jun.), pp. 9:1–9:33. ISSN: 1559-1131. doi: 10.1145/2180861.2180866. http://doi.acm.org/10.1145/2180861.2180866 .

AMICHAI-HAMBURGER, Y., HAYAT, T., 2017, "Social Networking". In: *The International Encyclopedia of Media Effects*, John Wiley & Sons, Inc. ISBN: 9781118783764. doi: 10.1002/9781118783764.wbieme0170. http://dx.doi.org/10.1002/9781118783764.wbieme0170 .

ANDREWS, S., TSOCHANTARIDIS, I., HOFMANN, T., 2003, "Support vector machines for multiple-instance learning". In: *Advances in Neural Information Processing Systems 15*, pp. 561–568. MIT Press.

ARORA, S., LIANG, Y., MA, T., 2017, "A Simple but Tough-to-Beat Baseline for Sentence Embeddings", .

ARTHUR, D., VASSILVITSKII, S., 2007, "K-means++: the advantages of careful seeding". In: *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*.

AU YEUNG, C.-M., GIBBINS, N., SHADBOLT, N., 2009, "User-induced links in collaborative tagging systems". pp. 787–796. doi: 10.1145/1645953.1646053. https://www.scopus.com/inward/record.uri?eid=2-s2.0-74549153304&doi=10.1145%2f1645953.1646053&partnerID=40&md5=9efd6f75550f47f6f2b07537d6869c2d . cited By.

BAUMEISTER, R. F., LEARY, M. R., 1997, "Writing narrative literature reviews". In: *Review of General Psychology*. http://psycnet.apa.org/doiLanding?doi=10.1037%2F1089-2680.1.3.311 .

BEM, D. J., 1995, "Writing a Review Article for Psychological Bulletin", *Psychological Bulletin*, pp. 172–177.

BENGIO, Y., DUCHARME, J., VINCENT, P., etal., 2003, "A Neural Probabilistic Language Model", *J. Mach. Learn. Res.*, v. 3 (mar.), pp. 1137–1155. ISSN: 1532-4435. http://dl.acm.org/citation.cfm?id=944919.944966 .

BENGIO, Y., LAMBLIN, P., POPOVICI, D., etal., 2007, "Greedy Layer-Wise Training of Deep Networks". In: Schölkopf, B., Platt, J. C., Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems 19*, MIT Press, pp. 153–160. http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf .

BENGIO, Y., COURVILLE, A. C., VINCENT, P., 2012, "Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives", *CoRR*, v. abs/1206.5538. http://arxiv.org/abs/1206.5538 .

BERGE, C., 1985, *Graphs and Hypergraphs*. Oxford, UK, UK, Elsevier Science Ltd. ISBN: 0720404797.

BERTIN-MAHIEUX, T., ECK, D., MAILLET, F., etal., 2008, "Autotagger: A Model for Predicting Social Tags from Acoustic Features on Large Music Databases", *Journal of New Music Research*, v. 37, n. 2, pp. 115–135. doi: 10.1080/09298210802479250. http://www.iro.umontreal.ca/~eckdoug/papers/2008_jnmr.pdf .

BLEI, D. M., NG, A. Y., JORDAN, M. I., 2003, "Latent Dirichlet Allocation", *J. Mach. Learn. Res.*, v. 3 (mar.), pp. 993–1022. ISSN: 1532-4435. http://dl.acm.org/citation.cfm?id=944919.944937 .

BORDES, A., USUNIER, N., GARCIA-DURAN, A., etal., 2013a, "Translating Embeddings for Modeling Multi-relational Data". In: Burges, C. J. C., Bottou, L., Welling, M., etal. (Eds.), *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pp. 2787–2795, a. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf .

BORDES, A., USUNIER, N., GARCIA-DURAN, A., etal., 2013b, "Translating Embeddings for Modeling Multi-relational Data". In: Burges, C.

J. C., Bottou, L., Welling, M., etal. (Eds.), *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pp. 2787–2795, b. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf .

BUCKLEY, C., VOORHEES, E. M., 2000, "Evaluating Evaluation Measure Stability". In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pp. 33–40, New York, NY, USA. ACM. ISBN: 1-58113-226-3. doi: 10.1145/345508.345543. http://doi.acm.org/10.1145/345508.345543 .

CATTUTO, C., BALDASSARRI, A., SERVEDIO, V. D. P., etal., 2007, "Vocabulary growth in collaborative tagging systems", *CoRR*, v. abs/0704.3316. http://arxiv.org/abs/0704.3316 .

CHARTE, F., RIVERA, A. J., DEL JESUS, M. J., etal., 2015, "QUINTA: A question tagging assistant to improve the answering ratio in electronic forums". In: *IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON)*, pp. 1–6, Sept. doi: 10.1109/EUROCON.2015.7313677.

CHEN, H.-M., CHANG, M.-H., CHANG, P.-C., etal., 2008, "SheepDog: Group and Tag Recommendation for Flickr Photos by Automatic Search-based Learning". In: *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pp. 737–740, New York, NY, USA. ACM. ISBN: 978-1-60558-303-7. doi: 10.1145/1459359.1459473. http://doi.acm.org/10.1145/1459359.1459473 .

CHIDLOVSKII, B., 2012, "Tag Ranking by Linear Relational Neighbourhood Propagation". In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 184–188, Aug. doi: 10.1109/ASONAM.2012.40.

CHOUBEY, R., 2011, *Tag recommendation using Latent Dirichlet Allocation.* Ph.D. Thesis, Kansas State University. http://krex.k-state.edu/dspace/handle/2097/9785 .

DATTOLO, A., FERRARA, F., TASSO, C., 2010, "The role of tags for recommendation: A survey". In: *3rd International Conference on Human System Interaction*, pp. 548–555, May. doi: 10.1109/HSI.2010.5514515.

DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., etal., 1990, "Indexing by latent semantic analysis", *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, v. 41, n. 6, pp. 391–407.

DIETTERICH, T. G., LATHROP, R. H., LOZANO-PÉREZ, T., 1997, "Solving the Multiple Instance Problem with Axis-parallel Rectangles", *Artif. Intell.*, v. 89, n. 1-2 (jan.), pp. 31–71. ISSN: 0004-3702. doi: 10.1016/S0004-3702(96)00034-3. http://dx.doi.org/10.1016/S0004-3702(96)00034-3 .

EDGAR, G., 2008, *Measure, Topology, and Fractal Geometry*. Springer-Verlag New York.

FLOECK, F., PUTZKE, J., STEINFELS, S., etal., 2010, "Imitation and quality of tags in social bookmarking systems - Collective intelligence leading to folksonomies", *Advances in Intelligent and Soft Computing*, v. 76, pp. 75–91. doi: 10.1007/978-3-642-14481-3_7. https://link.springer.com/chapter/10.1007%2F978-3-642-14481-3_7 . cited By.

GILLIS, N., 2014, "The Why and How of Nonnegative Matrix Factorization", *ArXiv e-prints*, (jan.).

GOH, D. H. L., LEE, C. S., CHUA, A. Y. K., etal., 2008, "An Examination of the Effectiveness of Social Tagging for Resource Discovery". In: *2008 International Workshop on Information-Explosion and Next Generation Search*, pp. 23–30, April. doi: 10.1109/INGS.2008.11.

GOLDER, S., HUBERMAN, B., 2006, "Usage patterns of collaborative tagging systems", *Journal of Information Science*, v. 32, n. 2, pp. 198–208. doi: 10.1177/0165551506062337. http://journals.sagepub.com/doi/pdf/10.1177/0165551506062337 . cited By.

GOLDER, S. A., HUBERMAN, B. A., 2005, "The Structure of Collaborative Tagging Systems", *CoRR*, v. abs/cs/0508082. http://arxiv.org/abs/cs/0508082 .

GONG, Y., ZHANG, Q., HUANG, X., 2017, "Hashtag recommendation for multimodal microblog posts", *Neurocomputing*. ISSN: 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2017.06.056. http://www.sciencedirect.com/science/article/pii/S0925231217311840 .

GUILLAUMIN, M., MENSINK, T., VERBEEK, J., etal., 2009, "TagProp: Discriminative metric learning in nearest neighbor models for image auto-

annotation". In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 309–316, Sept. doi: 10.1109/ICCV.2009.5459266.

HALPIN, H., ROBU, V., SHEPHERD, H., 2006, "The dynamics and semantics of collaborative tagging". v. 209. http://ceur-ws.org/Vol-209/saaw06-full01-halpin.pdf . cited By.

HAN, Y., WU, F., ZHUANG, Y., etal., 2010, "Multi-Label Transfer Learning With Sparse Representation", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 20, n. 8 (Aug), pp. 1110–1121. ISSN: 1051-8215. doi: 10.1109/TCSVT.2010.2057015.

HARALICK, R. M., SHAPIRO, L. G., 1985, "Image segmentation techniques", *Computer Vision, Graphics, and Image Processing*, v. 29, n. 1, pp. 100 – 132. ISSN: 0734-189X. doi: https://doi.org/10.1016/S0734-189X(85)90153-7. http://www.sciencedirect.com/science/article/pii/S0734189X85901537 .

HASSAN, M. T., KARIM, A., MANANDHAR, S., etal., 2009, "Discriminative Clustering for Content-based Tag Recommendation in Social Bookmarking Systems". In: *Proceedings of the 2009th International Conference on ECML PKDD Discovery Challenge - Volume 497*, ECMLPKDDDC'09, pp. 85–97, Aachen, Germany, Germany. CEUR-WS.org. http://ceur-ws.org/Vol-497/paper_24.pdf .

HEARST, M. A., 1994, "Multi-paragraph Segmentation of Expository Text". In: *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pp. 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics. doi: 10.3115/981732.981734. https://doi.org/10.3115/981732.981734 .

HELIC, D., KÖRNER, C., GRANITZER, M., etal., 2012, "Navigational Efficiency of Broad vs. Narrow Folksonomies". In: *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, HT '12, pp. 63–72, New York, NY, USA. ACM. ISBN: 978-1-4503-1335-3. doi: 10.1145/2309996.2310008. http://doi.acm.org/10.1145/2309996.2310008 .

HEYMANN, P., RAMAGE, D., GARCIA-MOLINA, H., 2008, "Social Tag Prediction". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 531–538, New York, NY, USA. ACM. ISBN: 978-1-60558-164-4. doi: 10.1145/1390334.1390425. http://doi.acm.org/10.1145/1390334.1390425 .

HU, J., WANG, B., LIU, Y., etal., 2012a, "Personalized Tag Recommendation Using Social Influence", *Journal of Computer Science and Technology*, v. 27, n. 3 (Jan), pp. 527–540. ISSN: 1860-4749. doi: 10.1007/s11390-012-1241-0. https://doi.org/10.1007/s11390-012-1241-0 .

HU, M., LIM, E. P., JIANG, J., 2010, "A Probabilistic Approach to Personalized Tag Recommendation". In: *2010 IEEE Second International Conference on Social Computing*, pp. 33–40, Aug. doi: 10.1109/SocialCom.2010.15. http://ieeexplore.ieee.org/document/5590886/ .

HU, R., HE, T., LI, F., etal., 2012b, "Tag recommendation based on tag-topic model". In: *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, v. 03, pp. 1501–1505, Octb. doi: 10.1109/CCIS.2012.6664635.

HUANG, J., SUN, H., HAN, J., etal., 2011, "Density-based shrinkage for revealing hierarchical and overlapping community structure in networks", v. 390 (06), pp. 2160–2171.

HUTTENLOCHER, D. P., KLANDERMAN, G. A., RUCKLIDGE, W. A., 1993, "Comparing Images Using the Hausdorff Distance", *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 15, n. 9 (sep.), pp. 850–863. ISSN: 0162-8828. doi: 10.1109/34.232073. https://doi.org/10.1109/34.232073 .

ILLIG, J., HOTHO, A., JÄSCHKE, R., etal., 2011, "A Comparison of Content-based Tag Recommendations in Folksonomy Systems". In: *Proceedings of the First International Conference on Knowledge Processing and Data Analysis*, KONT'07/KPP'07, pp. 136–149, Berlin, Heidelberg. Springer-Verlag. ISBN: 978-3-642-22139-2. http://dl.acm.org/citation.cfm?id=2022767.2022778 .

JAIN, A. K., 2010, "Data clustering: 50 years beyond K-means", *Pattern Recognition Letters*, v. 31, n. 8, pp. 651 – 666. ISSN: 0167-8655. doi: https://doi.org/10.1016/j.patrec.2009.09.011. http://www.sciencedirect.com/science/article/pii/S0167865509002323 . Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

JAIN, A. K., DUBES, R. C., 1988, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA, Prentice-Hall, Inc. ISBN: 0-13-022278-X.

JÄSCHKE, R., MARINHO, L., HOTHO, A., etal., 2007, "Tag Recommendations in Folksonomies". In: Kok, J. N., Koronacki, J., Lopez de Mantaras, R.,

etal. (Eds.), *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings*, pp. 506–514, Berlin, Heidelberg, Springer Berlin Heidelberg. ISBN: 978-3-540-74976-9. doi: 10.1007/978-3-540-74976-9_52. https://doi.org/10.1007/978-3-540-74976-9_52 .

JIN, Y., LI, R., LU, Z., etal., 2010, "Topic-Sensitive Tag Ranking". In: *2010 20th International Conference on Pattern Recognition*, pp. 629–632, Aug. doi: 10.1109/ICPR.2010.159.

JÚNIOR, E. A. C., MARINHO, V. Q., DOS SANTOS, L. B., 2017, "NILC-USP at SemEval-2017 Task 4: A Multi-view Ensemble for Twitter Sentiment Analysis", *CoRR*, v. abs/1704.02263. http://arxiv.org/abs/1704.02263 .

KAKADE, S. R., KAKADE, N. R., 2013, "A novel approach to link semantic gap between images and tags via probabilistic ranking". In: *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–6, Dec. doi: 10.1109/ICCIC.2013.6724166.

KATAKIS, I., TSOUMAKAS, G., VLAHAVAS, I., 2008, "Multilabel Text Classification for Automated Tag Suggestion". In: *Proceedings of the ECML/PKDD 2008 Discovery Challenge*. http://lpis.csd.auth.gr/publications/katakis_ecmlpkdd08_challenge.pdf .

KATARIA, S., AGARWAL, A., 2015a, "Distributed Representations for Content-Based and Personalized Tag Recommendation". In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1388–1395, Nova. doi: 10.1109/ICDMW.2015.240.

KATARIA, S., AGARWAL, A., 2015b, "Distributed Representations for Content-Based and Personalized Tag Recommendation". In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1388–1395, Novb. doi: 10.1109/ICDMW.2015.240.

KATARIA, S., 2016, "Recursive Neural Language Architecture for Tag Prediction", *CoRR*, v. abs/1603.07646. http://arxiv.org/abs/1603.07646 .

KAUFMAN, L., ROUSSEEUW, P. J., 1987. "Clustering by means of medoids". .

KISHIDA, K., 2005, "Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments", *NII Technical Reports*, v. 2005, n. 14 (9), pp. 1–19. ISSN: 1346-5597.

KRESTEL, R., FANKHAUSER, P., 2010, "Language Models and Topic Models for Personalizing Tag Recommendation". In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, v. 1, pp. 82–89, Aug. doi: 10.1109/WI-IAT.2010.29.

KULLBACK, S., LEIBLER, R. A., 1951, "On Information and Sufficiency", *Ann. Math. Statist.*, v. 22, n. 1 (03), pp. 79–86. doi: 10.1214/aoms/1177729694. https://doi.org/10.1214/aoms/1177729694 .

KÖRNER, C., BENZ, D., HOTHO, A., etal., 2010, "Stop thinking, start tagging: Tag semantics emerge from collaborative verbosity". In: *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pp. 521–530, 01.

LE, Q., MIKOLOV, T., 2014, "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1188–II–1196. JMLR.org. http://dl.acm.org/citation.cfm?id=3044805.3045025 .

LEGINUS, M., DOLOG, P., ŽEMAITIS, V., 2012, "Improving Tensor Based Recommenders with Clustering". In: Masthoff, J., Mobasher, B., Desmarais, M. C., etal. (Eds.), *User Modeling, Adaptation, and Personalization: 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012. Proceedings*, pp. 151–163, Berlin, Heidelberg, Springer Berlin Heidelberg. ISBN: 978-3-642-31454-4. doi: 10.1007/978-3-642-31454-4_13. https://doi.org/10.1007/978-3-642-31454-4_13 .

LI, X., SNOEK, C. G. M., WORRING, M., 2009, "Learning Social Tag Relevance by Neighbor Voting", *IEEE Transactions on Multimedia*, v. 11, n. 7 (Nov), pp. 1310–1322. ISSN: 1520-9210. doi: 10.1109/TMM.2009.2030598.

LI, X., WANG, H., GU, B., etal., 2015, "Data Sparseness in Linear SVM". In: *IJCAI*.

LIU, S., ZHU, Y., GUO, J., etal., 2013, "A Blending Method for Automated Social Tagging". In: *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, v. 1, pp. 115–120, Nov. doi: 10.1109/WI-IAT.2013.17.

MACQUEEN, J., 1967, "Some methods for classification and analysis of multivariate observations". In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

MARINHO, L. B., HOTHO, A., JSCHKE, R., etal., 2012, *Recommender Systems for Social Tagging Systems*. Springer Publishing Company, Incorporated. ISBN: 1461418933, 9781461418931.

MARLOW, C., NAAMAN, M., BOYD, D., etal., 2006, "HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, to Read". In: *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, HYPERTEXT '06, pp. 31–40, New York, NY, USA. ACM. ISBN: 1-59593-417-0. doi: 10.1145/1149941.1149949. http://doi.acm.org/10.1145/1149941.1149949 .

MARON, O., RATAN, A. L., 1998, "Multiple-Instance Learning for Natural Scene Classification". In: *In The Fifteenth International Conference on Machine Learning*, pp. 341–349. Morgan Kaufmann.

MARTÍNEZ, E., CELMA, O., SORDO, M., etal., 2009, "Extending the folksonomies of freesound.org using content-based audio analysis". pp. 65–70. http://mtg.upf.edu/files/publications/SMC09_emartinez_ocelma_msordo_bdejong_xserra.pdf . cited By.

MATHES, A., 2004, "Folksonomies - cooperative classification and communication through shared metadata", *Computer Mediated Communication*, (December). http://adammathes.com/academic/computer-mediated-communication/folksonomies.html .

MIKA, P., 2007, "Ontologies are us: A unified model of social networks and semantics", *Web Semantics: Science, Services and Agents on the World Wide Web*, v. 5, n. 1, pp. 5 – 15. ISSN: 1570-8268. doi: http://dx.doi.org/10.1016/j.websem.2006.11.002. http://www.sciencedirect.com/science/article/pii/S1570826806000552 . Selected Papers from the International Semantic Web Conference.

MIKOLOV, T., SUTSKEVER, I., CHEN, K., etal., 2013, "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pp. 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality .

MIKOLOV, T., CHEN, K., CORRADO, G., etal., 2013b, "Efficient Estimation of Word Representations in Vector Space", *CoRR*, v. abs/1301.3781. http://arxiv.org/abs/1301.3781 .

MOXLEY, E., MEI, T., HUA, X. S., etal., 2008, "Automatic video annotation through search and mining". In: *2008 IEEE International Conference on Multimedia and Expo*, pp. 685–688, June. doi: 10.1109/ICME.2008.4607527.

MROSEK, J., BUSSMANN, S., ALBERS, H., etal., 2009, "Content- and Graph-based Tag Recommendation: Two Variations", https://www.kde.cs.uni-kassel.de/ws/dc09/papers/paper_18.pdf .

NIKOLOPOULOS, S., CHATZILARI, E., GIANNAKIDOU, E., etal., 2009, "Towards fully un-supervised methods for generating object detection classifiers using social data". In: *2009 10th Workshop on Image Analysis for Multimedia Interactive Services*, pp. 230–233, May. doi: 10.1109/WIAMIS.2009.5031475.

OBAR, J. A., WILDMAN, S., 2015, "Social media definition and the governance challenge: An introduction to the special issue", *Telecommunications Policy*, v. 39, n. 9, pp. 745 – 750. ISSN: 0308-5961. doi: https://doi.org/10.1016/j.telpol.2015.07.014. http://www.sciencedirect.com/science/article/pii/S0308596115001172 . SPECIAL ISSUE ON THE GOVERNANCE OF SOCIAL MEDIA.

PAGE, L., BRIN, S., MOTWANI, R., etal., 1999, *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66, Stanford InfoLab, November. http://ilpubs.stanford.edu:8090/422/ . Previous number = SIDL-WP-1999-0120.

PAN, S. J., YANG, Q., 2010, "A Survey on Transfer Learning", *IEEE Trans. on Knowl. and Data Eng.*, v. 22, n. 10 (oct.), pp. 1345–1359. ISSN: 1041-4347. doi: 10.1109/TKDE.2009.191. http://dx.doi.org/10.1109/TKDE.2009.191 .

PERALTA, V., 2007, "Extraction and Integration of MovieLens and IMDb Data", (08).

PETERS, I., 2009, *Folksonomies. Indexing and Retrieval in Web 2.0.* 1st ed. Hawthorne, NJ, USA, Walter de Gruyter & Co. ISBN: 3598251793, 9783598251795.

RABINER, L. R., 1990, "Readings in Speech Recognition". Morgan Kaufmann Publishers Inc., cap. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296, San Francisco, CA, USA. ISBN: 1-55860-124-4. http://dl.acm.org/citation.cfm?id=108235.108253 .

RAWASHDEH, M., KIM, H.-N., ALJA'AM, J. M., etal., 2013, "Folksonomy link prediction based on a tripartite graph for tag recommendation", *Journal of Intelligent Information Systems*, v. 40, n. 2 (Apr), pp. 307–325. ISSN: 1573-7675. doi: 10.1007/s10844-012-0227-2. https://doi.org/10.1007/s10844-012-0227-2 .

RAWAT, W., WANG, Z., 2017, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", *Neural Computation*, v. 29, n. 9, pp. 2352–2449. doi: 10.1162/neco_a_00990. https://doi.org/10.1162/neco_a_00990 . PMID: 28599112.

RENDLE, S., SCHMIDT-THIEME, L., 2009, "Factor Models for Tag Recommendation in Bibsonomy". In: *Proceedings of the 2009th International Conference on ECML PKDD Discovery Challenge - Volume 497*, ECMLPKDDDC'09, pp. 235–242, Aachen, Germany, Germany. CEUR-WS.org. https://www.kde.cs.uni-kassel.de/ws/dc09/papers/paper_13.pdf .

RENDLE, S., BALBY MARINHO, L., NANOPOULOS, A., etal., 2009, "Learning Optimal Ranking with Tensor Factorization for Tag Recommendation". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pp. 727–736, New York, NY, USA. ACM. ISBN: 978-1-60558-495-9. doi: 10.1145/1557019.1557100. http://doi.acm.org/10.1145/1557019.1557100 .

RIFKIN, R., KLAUTAU, A., 2004, "In Defense of One-Vs-All Classification", *J. Mach. Learn. Res.*, v. 5 (dec.), pp. 101–141. ISSN: 1532-4435. http://dl.acm.org/citation.cfm?id=1005332.1005336 .

SATTIGERI, P., THIAGARAJAN, J. J., SHAH, M., etal., 2014, "A scalable feature learning and tag prediction framework for natural environment sounds". In: *2014 48th Asilomar Conference on Signals, Systems and Computers*, pp. 1779–1783, Nov. doi: 10.1109/ACSSC.2014.7094773.

SCHIFANELLA, R., BARRAT, A., CATTUTO, C., etal., 2010, "Folks in Folksonomies: Social Link Prediction from Shared Metadata", *CoRR*, v. abs/1003.2281. http://arxiv.org/abs/1003.2281 .

SHEN, C., JIAO, J., YANG, Y., etal., 2009, "Multi-instance multi-label learning for automatic tag recommendation". In: *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 4910–4914, Oct. doi: 10.1109/ICSMC.2009.5346261.

SI, X., SUN, M., 2010, "Tag Allocation Model: Model Noisy Social Annotations by Reason Finding". In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, v. 1, pp. 413–416, Aug. doi: 10.1109/WI-IAT.2010.85.

SI, X., SUN, M., 2008, "Tag-LDA for Scalable Real-time Tag Recommendation", v. 6 (11). https://www.yumpu.com/en/document/view/40719286/tag-lda-for-scalable-real-time-tag-recommendation .

SI, X., LIU, Z., LI, P., etal., 2009, "Content-based and Graph-based Tag Suggestion". In: *Proceedings of the 2009th International Conference on ECML PKDD Discovery Challenge - Volume 497*, ECMLPKDDDC'09, pp. 243–260, Aachen, Germany, Germany. CEUR-WS.org. https://www.kde.cs.uni-kassel.de/ws/dc09/results/papers/paper_14.pdf .

SOKOLOVA, M., LAPALME, G., 2009, "A systematic analysis of performance measures for classification tasks", *Information Processing & Management*, v. 45, n. 4, pp. 427 – 437. ISSN: 0306-4573. doi: https://doi.org/10.1016/j.ipm.2009.03.002. http://www.sciencedirect.com/science/article/pii/S0306457309000259 .

SONG, Y., ZHUANG, Z., LI, H., etal., 2008, "Real-time Automatic Tag Recommendation". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 515–522, New York, NY, USA. ACM. ISBN: 978-1-60558-164-4. doi: 10.1145/1390334.1390423. http://doi.acm.org/10.1145/1390334.1390423 .

SONG, Y., ZHANG, L., GILES, C. L., 2011, "Automatic Tag Recommendation Algorithms for Social Recommender Systems", *ACM Trans. Web*, v. 5, n. 1 (feb.), pp. 4:1–4:31. ISSN: 1559-1131. doi: 10.1145/1921591.1921595. http://doi.acm.org/10.1145/1921591.1921595 .

SORDO, M., LAURIER, C., CELMA, Ò., 2007, "Annotating Music Collections How content-based similarity helps to propagate labels". In: *8th International Conference on Music Information Retrieval*, Vienna, Austria. http://mtg.upf.edu/files/publications/7c086c-ISMIR-2007-msordo-claurier.pdf .

SYMEONIDIS, P., NANOPOULOS, A., MANOLOPOULOS, Y., 2008, "Tag recommendations based on tensor dimensionality reduction". pp. 43–50. doi: 10.1145/1454008.1454017. http://dl.acm.org/ft_gateway.cfm?id=1454017&type=pdf&CFID=975119194&CFTOKEN=74174692 . cited By.

TAO, R., YAO, T., 2016, "Tag recommendation based on Paragraph Vector". In: *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2786–2789, Oct. doi: 10.1109/CompComm.2016.7925205.

TRABELSI, C., MOULAHI, B., YAHIA, S., 2012, "HMM-CARe: Hidden Markov models for context-aware tag recommendation in folksonomies". pp. 957–961. doi: 10.1145/2245276.2245461. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84863575351&doi=10.1145%2f2245276.2245461&partnerID=40&md5=c1941e7f351c3dba08f4ff967543524b . cited By.

TSOUMAKAS, G., KATAKIS, I., 2007, "Multi-label classification: An overview", *Int J Data Warehousing and Mining*, v. 2007, pp. 1–13.

TSOUMAKAS, G., KATAKIS, I., VLAHAVAS, I., 2010, "Mining Multi-label Data". In: Maimon, O., Rokach, L. (Eds.), *Data Mining and Knowledge Discovery Handbook*, pp. 667–685, Boston, MA, Springer US. ISBN: 978-0-387-09823-4. doi: 10.1007/978-0-387-09823-4_34. https://doi.org/10.1007/978-0-387-09823-4_34 .

VAN DER MAATEN, L., POSTMA, E. O., VAN DEN HERIK, H. J., 2008. "Dimensionality Reduction: A Comparative Review". .

VAN LEEUWEN, M., PUSPITANINGRUM, D., 2012, "Improving tag recommendation using few associations", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 7619 LNCS, pp. 184–194. doi: 10.1007/978-3-642-34156-4_18. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84868032350&doi=10.1007%2f978-3-642-34156-4_18&partnerID=40&md5=9207f41d00f57a8bec31e43559703813 . cited By.

WAL, V. D., 2005a. "Explaining and showing broad and narrow folksonomies". a. http://www.vanderwal.net/random/entrysel.php?blog=1635 .

WAL, V. D., 2005b. "Folksonomy Explanations". b. http://www.vanderwal.net/random/entrysel.php?blog=1622 .

WANG, J., PENG, J., LIU, O., 2015, "A classification approach for less popular webpages based on latent semantic analysis and rough set model", *Expert Systems with Applications*, v. 42, n. 1, pp. 642 – 648. ISSN: 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2014.08.013. http://www.sciencedirect.com/science/article/pii/S0957417414004898 . cited By.

WEI HSU, C., CHUNG CHANG, C., JEN LIN, C., 2010. "A practical guide to support vector classification". .

WIETING, J., BANSAL, M., GIMPEL, K., etal., 2015, "Towards Universal Paraphrastic Sentence Embeddings", *CoRR*, v. abs/1511.08198. http://arxiv.org/abs/1511.08198 .

WOHLIN, C., 2014, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pp. 38:1–38:10, New York, NY, USA. ACM. ISBN: 978-1-4503-2476-2. doi: 10.1145/2601248.2601268. http://doi.acm.org/10.1145/2601248.2601268 .

WOLD, S., ESBENSEN, K., GELADI, P., 1987, "Principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, v. 2, n. 1, pp. 37 – 52. ISSN: 0169-7439. doi: https://doi.org/10.1016/0169-7439(87)80084-9. http://www.sciencedirect.com/science/article/pii/0169743987800849 . Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

WU, Y., YAO, Y., XU, F., etal., 2016, "Tag2Word: Using Tags to Generate Words for Content Based Tag Recommendation". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pp. 2287–2292, New York, NY, USA. ACM. ISBN: 978-1-4503-4073-1. doi: 10.1145/2983323.2983682. http://doi.acm.org/10.1145/2983323.2983682 .

ZHANG, J., LIU, X., ZHUO, L., etal., 2015, "Social images tag ranking based on visual words in compressed domain", *Neurocomputing*, v. 153, pp. 278 – 285. ISSN: 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2014.11.027. http://www.sciencedirect.com/science/article/pii/S0925231214015604 .

ZHANG, M.-L., ZHOU, Z.-H., 2007, "ML-KNN: A Lazy Learning Approach to Multi-label Learning", *Pattern Recogn.*, v. 40, n. 7 (jul.), pp. 2038–2048.

ISSN: 0031-3203. doi: 10.1016/j.patcog.2006.12.019. http://dx.doi.org/10.1016/j.patcog.2006.12.019 .

ZHANG, M.-L., ZHOU, Z.-H., 2006, "Multi-Instance Multi-Label Learning with Application to Scene Classification". In: Schölkopf, B., Platt, J. C., Hoffman, T. (Eds.), *Advances in Neural Information Processing Systems 19*, MIT Press, pp. 1609–1616. http://papers.nips.cc/paper/3047-multi-instance-multi-label-learning-with-application-to-scene-classification.pdf .

ZHANG, Y., YI, D., WEI, B., etal., 2014, "A GPU-accelerated non-negative sparse latent semantic analysis algorithm for social tagging data", *Information Sciences*, v. 281, pp. 687 – 702. ISSN: 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2014.04.047. http://www.sciencedirect.com/science/article/pii/S002002551400512X . Multimedia Modeling.

ZHAO, W., GUAN, Z., LIU, Z., 2015, "Ranking on heterogeneous manifolds for tag recommendation in social tagging services", *Neurocomputing*, v. 148, pp. 521–534. ISSN: 0925-2312. doi: 10.1016/j.neucom.2014.07.011. http://www.sciencedirect.com/science/article/pii/S0925231214008893 . cited By.

ZUBIAGA, A., GARCÍA-PLAZA, A. P., FRESNO, V., etal., 2009, "Content-Based Clustering for Tag Cloud Visualization". In: *2009 International Conference on Advances in Social Network Analysis and Mining*, pp. 316–319, July. doi: 10.1109/ASONAM.2009.19.

Appendix A

Code Layout

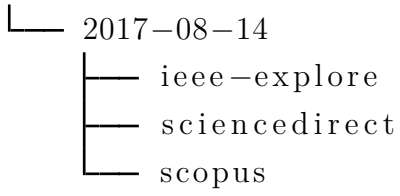
The project as a whole was based off a project template called *Cookiecutter Data-science*[1] which is recommended for aiding reproducibility in data science projects.

The project[2] is organized as follows: .

```
social-tags/
├── data/                    <-- saved data
│   ├── external
│   ├── interim
│   ├── processed
│   └── raw
│
├── models/                  <-- trained models
│
├── notebooks/               <-- all code for training all models
│   ├── helper-scripts/
│   ├── delicious-t140/
│   └── movielens-imdb/
│
├── scripts/                 <-- other scripts
│
├── src/                     <-- utils and helper code
│   ├── data/                <-- code for building the datasets
│   ├── features/            <-- code for extracting features
│   ├── helpers/             <-- helper code
│   └── utils/               <-- helper code that's fit for reuse
│
└── systematic-search/       <-- evidence of collected material
```

[1] Available online at https://drivendata.github.io/cookiecutter-data-science/

[2] Available online at https://github.com/queirozfcom/auto-tagger/tree/master/social-tags


2017-08-14
ieee-explore
sciencedirect
scopus