Submission date: 14/10/2019
Resubmission date: 16/03/2020
Camera ready submission date: 25/05/2020

1st round notification: 23/12/2019
2nd round notification: 12/05/2020
Available online: 29/05/2020
Publication date: 31/07/2020

Section: extended version
Extended versions of papers presented at the VIII Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2019)

# A Graph-Based Method for Predicting the Helpfulness of Apps Opinions

**Rogério Figueredo de Sousa[1], Rafael Torres Anchiêta[1], Maria das Graças Volpe Nunes[1]**

[1]Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Sciences, University of São Paulo
Av. Trabalhador São Carlense, 400 – 13.566-590 – São Carlos – SP – Brazil

{rogerfig, rta}@usp.br, gracan@icmc.usp.br

***Abstract.*** *This paper presents a new approach to predict the helpfulness of opinions. Usually, researchers in this area use tables of attribute-value to aggregate the features that represent the evaluated texts. Although that representation is common, it considers that the objects are independent. We argue that among the discriminant factors of the helpfulness of opinions, there are dependent factors of the relationship among the opinion-forming elements. Thus, we modeled this task as a network, considering the information of relations among objects in the network (comments, stars, and words). A regularization technique of graphs is used to extract the relevant features of graph structure and, after that, the comments are classified as helpful or unhelpful. We compared our network model with two baselines methods, one based on fuzzy logic and another based on Neural Networks. Our model outperformed the fuzzy logic and Neutal Network methods in 0.17 and 0.19 of F-measure, respectively. The main advantages of our approach are that few data are necessary to helpfulness classification and the relationships may help in the understanding the classification, explaining the reasons for a determinate classification.*

***Keywords.*** *Natural Language Processing; Helpfulness Prediction; Opinion Mining.*

## 1. Introduction

Choosing a product to buy or a movie to watch is today one of the most frequent activities of Internet users. A common practice for these people is to search for information about products or services for which they are interested in specialized websites. In addition to technical information, it is the opinions of other users that interest them the most. If the advertiser is only interested in revealing the qualities of the product, the user who has had a good or bad experience with the product is only interested in contributing to other consumers. According to [Liu 2012], opinions are central to most human activities and are capable of influencing human behavior.

User-generated content (UGC) is a major source of web content, and product and service comments form a large portion of that content. Not every comment (or opinion

Cite as: Sousa, R. F., Anchiêta, R. T. & Nunes, M. G. V. (2020). A Graph-Based Method for Predicting the Helpfulness of Apps Opinions. iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 13(4), 06-21. DOI: 10.5753/isys.2020.821

or review) is considered useful or relevant by other users. Indeed, some of this content may be considered unwanted [Kim et al. 2006]. Mentions that unwanted content includes poorly written texts, vague opinions, texts with questionable content, etc. That is, user-generated content varies greatly in quality and such texts do not help readers' decision making.

Another factor to difficult decision making is the huge amount of comments available on the web, making finding relevant content even more complicated. Moreover, it is impossible for users to read all the good quality content available. Deciding whether a comment carries potentially useful information for decision making is the central problem of this paper.

In Natural Language Processing (NLP), the Opinion Helpfulness Prediction task comprises the definition of models for characterizing good quality content and the proposition of methods for classifying opinions on helpfulness. Identifying relevant (helpful) content in user comments can help other users in decision making as well as support other NLP processes, such as the opinion summarization [Anchiêta et al. 2017].

The e-commerce sites' concern with presenting helpfulness content is great, which is why some of them ask for explicit feedback from the user: is this comment helpful or not? Thus, the comments presented are sorted according to the votes they have received, the most helpful first. Some problems arise from this form or manual voting [Kim et al. 2006, Liu et al. 2007, Singh et al. 2017]:

1. Helpful new comments will hardly be at the top of the ranking. It takes some time for several people to vote and the comment to gain proper visibility;
2. Items that have low visitor traffic will not have enough votes to generate a reliable ranking;
3. People may make a false assessment of the helpfulness of comments. Spammers are the ones who can dishonestly evaluate some comments so that they go up or down the ranking.

To avoid this type of problem, it is necessary to learn existing features in rankings of already consolidated comments and thus automatically evaluate the helpfulness of comments generated by users. The vast majority of known works perform this task using the attribute-value representation. But, despite being widely used, it is not able to capture relationship information between objects. The data structures that best represent relationships between objects are networks.

In this work, the task of helpfulness prediction was modeled as a heterogeneous network. To evaluate this approach, we used the UTLCorpus[Sousa et al. 2019], a recently released corpus, specifically the Google Play Store sub-corpus and, then, compared our approach with a well-known baseline based on fuzzy logic [de Sousa et al. 2015] and its evolution based on Neural Networks (NN)[Santos et al. 2016]. The network-based approach exceeded the fuzzy baseline by 0.17 points in F1 measure and 0.19 points in F1 measure on NN baseline, showing that our approach is feasible to predict whether a comment is helpful or not.

It is important to highlight that, as far as we know, this is the first work that models the helpfulness prediction task as a heterogeneous network. In addition, the approaches

are applied and evaluated in comments written in Portuguese in order to foster research in this area for this language.

The rest of the paper is organized as follows. In Section 2, the main related works are briefly described. In Section 3, we present the corpus and the developed modeling, as well as the steps to predict the helpfulness of the review. In Section 4, the performed experiments are detailed. Finally, Section 5 concludes the work, presenting future directions.

## 2. Related Work

[Zeng et al. 2014] attempted to include the sentiment polarity on binary helpfulness classification task. They model the problem in three classes. In the first class are the Helpful positive reviews (star rating $\epsilon$ [4,5] and helpfulness score $h > threshold$). In second class, are the Helpful negative reviews (star rating $\epsilon$ [1,2] and helpfulness score $h > threshold$), and in the last class, are the Unhelpful reviews (helpfulness score $h < threshold$). The purpose of this class division is to assess the impact of sentiment polarity on the helpfulness prediction task. Their dataset contains $8,690$ reviews from Amazon.com. The helpfulness score threshold was set empirically. The best result reported was 72.82% of accuracy on ten-fold cross-validation and specifically, for each class, the results reported were 69% on macro-f1 for the helpful positives; 79.5% on macro-f1 for the helpful negatives and 80% on macro-f1 for the unhelpful ones.

[Krishnamoorthy 2015] builds a model for binary helpfulness classification task. The authors introduced some linguistic features as features of their model. These features was based on a model named Linguistic Category Model (LCM) [Semin 2011] and these features are capable to identify the emotional state of the reviewer. They assume that linguistic categories are perceived by consumers and impact their vote behavior and, hence, the helpfulness of a review. For experimentation, they used an extracted corpus from Amazon.com (MDSD) and, using a threshold $h = 0.60$, the authors build three machine learning methods for helpfulness binary classification. Among the built models, the best result was achieved by an Random Forest model, reaching 84% of F-measure using all proposed features. The LCM features achieved the best result among other features.

In [Malik and Hussain 2017] the authors treated the utility prediction as a sorting task. They devised a method for calculating the emotion score of comments considering some specific feelings such as confidence, surprise, anger, sadness, etc. They used these scores as a feature in addition to more general ones, such as product ranking on Amazon, product price, number of verbs, nouns, adjectives and adverbs, among others. They modeled a Deep Neural Network and also evaluated the method on an Amazon.com subcorpus. The authors reported results on average of 89% F1 using positive emotions and 87% F1 using negative emotions.

In addition to the work described above, it is worth mentioning the work done focusing on the Portuguese language. Four of them will be described bellow.

The work of [Martins and Tacla 2015] presents a methodology focused on identifying features that have the greatest influence on utility votes. Experiments are applied to service domain (hotel) reviews. The authors propose several features that are able to

characterize comments, and they are divided into two categories: textual and semantic. Textual features consist mainly of intelligibility metrics. For their extraction, they use an adapted version of Coh-Metrix-Port [Scarton and Aluísio 2010]. In addition to the intelligibility index, other textual metrics, such as number of sentences, words and syllables, are used. For semantic features, the LSA (Latent Semantic Analysis) [Landauer et al. 1998] is used. The results of this work confirm the positive impact of semantic features in evaluating the helpfulness of opinions also for the Portuguese language. The intelligibility index revealed that longer and more complex opinions are more helpful than shorter and more intelligible opinions.

A different approach to classifying the importance of comments for the Portuguese language was presented in [de Sousa et al. 2015]. The authors proposed a Fuzzy Inference System to classify product domain comments into 4 classes: Insufficient, Sufficient, Good, and Excellent through 3 features: author reputation, number of type pairs (feature, opinion word), and richness of vocabulary. Comments are ranked according to the value expressed by the system. After sorting the list, several cut points were defined successively, and at each cut point a baseline method was applied to define the polarity of the comments. The authors compared the results on the subsets by applying the same method to the complete set. The results showed that a cut-off point considering only 10% of the comments obtained a better result than the complete set analysis. The authors presented a 10% increase in f-measure for positive comments and about 20% f-measure for negative comments.

The work of [Santos et al. 2016] has extended the work of [de Sousa et al. 2015]. They improved the definitions of some characteristics and proposed an experimental study to compare the previous approach of fuzzy systems with the use of Artificial Neural Networks. Two topologies of artificial neural networks were proposed. The authors reported that they could not improve the previous results, but argued that this is due to some factors: the samples obtained scattered expected results, which made the generalization of the network difficult; and none of the candidate topologies reached the minimum accuracy. The minimum accuracy serves as the minimum limit to consider the trained network. 52.48% of f-measure was achieved for positive comments and 62.53% of f-measure for negative comments.

Finally, [Barbosa and Moura 2016] assessed the helpfulness of opinions in the field of games. The authors collected comments from Steam[1] and used the authoring features of the opinions, textual characteristics and metadata existing on the site as input for an artificial neural network of the MLP (Multi-layer Perceptron) type to infer the helpfulness of the reviews. After the experiments, they reported good results and showed that the metrics related to authorship were more relevant along with the size of the text. On the other hand, the date of posting of the comments did not have a strong impact on the evaluation.

Looking at the involved elements in the scenario of users opinions - the opinion text, the reviewer, the object of the opinion, the reader's reaction, the reader, other opinions on the same topic, etc. - one can realize some relationships among them. In order

[1]http://store.steampowered.com

to verify the relevance of these relations for the task of determining the helpfulness of an opinion, in the next sections, we discuss the use of networks to model this task.

## 3. Helpfulness Prediction

### 3.1. Corpus

In our previous work [de Sousa et al. 2019], we used a small corpus collected from Google Play Store with $2,000$ reviews from $10$ apps of the communication category (see Table 1). But, recently a new corpus has been made available. Therefore, to evaluate our modeling, we used the UTLCorpus corpus presented in [Sousa et al. 2019].

Table 1. Number of applications and comments extracted in previous work

| App | # Comments | App | # Comments |
|---|---|---|---|
| Facebook | 200 | Skype | 200 |
| Google Allo | 200 | Snapchat | 200 |
| Hangouts | 200 | Telegram | 200 |
| Mensagens | 200 | Viber | 200 |
| Messenger | 200 | WhatsApp | 200 |
| **Total** | 2,000 | | |

The UTLCorpus contains $2,881,589$ reviews ($1,839,851$ of movies and $1,041,738$ of apps). For this work, we use only the apps domain. The creators of the UTLCorpus collected the apps reviews by crawling the Google Play Store[2]. They gathered reviews from $243$ apps and after the removal of the irrelevant ones, the final corpus was left with $921,257$ reviews. Table 2 presents some detailed information about UTLCorpus and a comparison between our previous corpus[de Sousa et al. 2019] and the UTLCorpus. That table shows the number of reviews for each class with the applied method. First, the reviews are grouped by their categories. After grouping, we consider only the votes of the comments. Next, we sort the votes in descending order. Then, we consider only the number of votes greater than one. Finally, all reviews with more votes than the first percentile of the distribution are considered helpful. On the other hand, all comments with fewer votes than the previous threshold and which were collected at least five days after their publication are considered non-helpful.

It is important to highlight that the UTLCorpus have apps of several categories which is different from the approach of our previous work [de Sousa et al. 2019]. In that past work, one assumption was that using reviews from the same category would help the prediction task because are believed to have similar terms and/or topics [Anchiêta and Moura 2017]. However, it is important to evaluate our approach with reviews from many different categories. This is one reason for the use of UTLCorpus, besides its size.

For the best of our knowledge, there is only more one available corpus in Brazilian Portuguese that contains information about helpfulness [Hartmann et al. 2014], however, the domain is different from that proposed here.

[2]https://play.google.com/store

Table 2. Comparison between the corpora

| | **Previous Corpus** | **UTLCorpus** |
|---|---|---|
| **# documents** | $2,000$ | $921,257$ |
| **# types** | $6,421$ | $419,713$ |
| **# tokens** | $33,749$ | $11,919,636$ |
| **# Helpful Reviews** | $800$ (40%) | $50,166$ (5%) |
| **# Not Helpful Reviews** | $800$ (40%) | $871,091$ (95%) |

A comment on Google Play contains the comment text itself, the comment's author, the number of stars, the comment's date and the number of likes the comment received (see Figure 1). The number of stars and the number of likes are in the range of $[0, 5]$ and $[0, +\infty)$, respectively. The source code and the previous dataset are still available upon request by email.

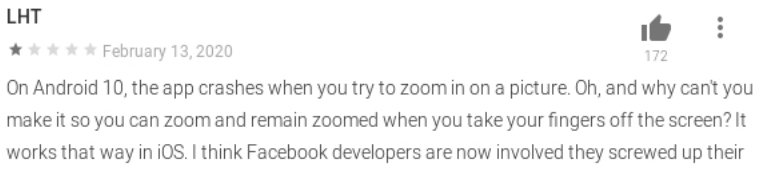



Figure 1. Example of a Google Play comment

The helpfulness votes are often used to define which comments are helpful or not, by calculating the helpfulness score ($h \ \epsilon \ [0, 1]$) using the Equation 1. According to [Diaz and Ng 2018], the task mainly include score regression, binary review classification, and review ranking. For binary classification, a threshold could be applied to helpfulness score to split the reviews on helpful or not.

$$h = \frac{helpful\ votes}{helpful\ votes + unhelpful\ votes} \tag{1}$$

$$utility_{c_i} = \frac{votes_{c_i}}{\sum_{c_j \epsilon C} c_j} \tag{2}$$

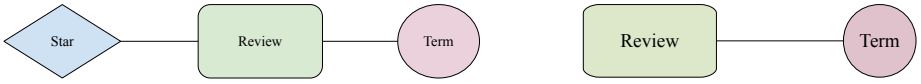
In the Google Play Store, the reviews do not have the "thumbs down" values. For that reason, we had to do an adaptation. Our first idea was applied on a previous paper [de Sousa et al. 2019]. We use the Equation 2 to assign a utility score for each review, where the utility score of the review $i$ is equal to the number of votes of the review $i$ divided by the sum of all votes from reviews ($\sum_{c_j \epsilon C} c_j$ where $C$ is the set of reviews excluding the review $i$). After calculating the score, we sorted the reviews in descending order, considering the first 40% reviews as helpful, and the last 40% reviews as not helpful. The remaining 20% (middle of the list) are disregarded, as they may be noisy, presenting overlap between labels. Therefore, from 2,000 reviews, it has left 1,600

reviews (800 helpful and 800 not helpful). This approach works well for small number of reviews. However, this approach does not work on UTLCorpus, as there are many reviews with no helpfulness votes, and few reviews with many votes. Thus, it is not interesting to split the data set into two equal parts. Fortunately, the authors of the UTLCorpus presented a solution for this issue.

### 3.2. Proposed modeling

The helpfulness prediction task has been seen as a regression, classification or ranking problem, modeled as an attribute-value table. However, in this work, this task was modeled as a heterogeneous network. We propose 4 modeling variations. In Figure 2, the scheme of the modeled networks is shown, where `Star`, `Review`[3], and `Term` are the network nodes. The first model is identical with the depicted in Figure 2a and it has a variation considering weights in edges between `Review` and `Term` nodes. Another model is identical with 2b and it has a weighted variation.

Figure 2. Scheme of heterogeneous network for helpfulness prediction




(a) Star-Reviews-Term network (b) Reviews-Term network

In order to instantiate the heterogeneous network, we developed a methodology organized in 4 steps from the extracted comments. In Figure 3, an overview of the methodology is presented.

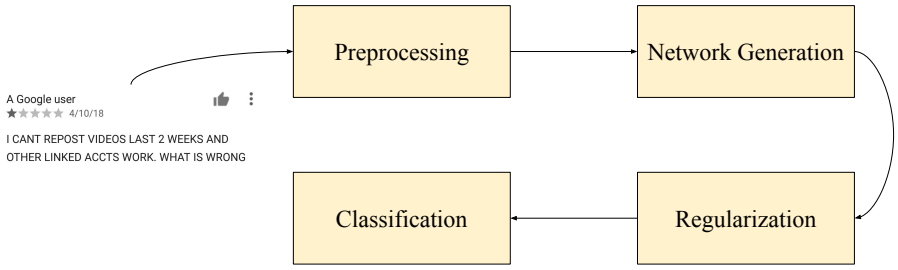



Figure 3. Methodology to instantiate the heterogeneous network

In the first step, the comments were pre-processed (subsection 3.3). Then, the heterogeneous network was generated (subsection 3.4). After that, a regularization algorithm was applied over the network (subsection 3.5). Finally, the comments were classified into helpful or not helpful (subsection 3.6).

It is important to point out that the approach as a whole is language independent. The only step that makes use of linguistic components is pre-processing, but it is possible to replace the linguistic resources so that the whole process is applied to another language.

[3]In this paper, we used the words review and comment on the interchangeable way.

### 3.3. Pre-processing

At this stage, the users' comments were pre-processed in order to be normalized, the most relevant terms of the comments were selected, and the number of stars attributed to each comment were extracted.

Google Play comments are similar to tweets: they are usually short and do not follow grammar and punctuation rules. Thus, a textual normalizer developed for Portuguese [Bertaglia and Nunes 2016] was applied to each comment of the corpus to minimize the amount of textual noise that could hinder further processing steps. These noises include misspellings, abbreviations, internet slangs, repetitions, etc. NLPnet's Part-Of-Speech Tagger [Fonseca and Rosa 2013] was then used to extract open-class words such as noun, verb, adjective, and adverb. Finally, to obtain the stems of each comment word, a stemming algorithm was applied [Orengo and Huyck 2001].

### 3.4. Network generation

The process of generating a network is simple. After the pre-processing step, we created nodes of the types `star`, `review`, and `term`. We present the guidelines to create each network-type below.

- Network Type Star-Review-Term (SRT): The `review` and `star` nodes are linked to each other by an edge according to the number of stars of the review. We adopted this same strategy to link the `review` and `term` nodes, whenever the term is present in the review. The `star` nodes neither can be linked to each other nor to the `term` node. In the same way, the `term` node can not be linked to each other. The network is undirected and in the unweighted variation (USRT) there is no edges with weights. In Figure 4, we illustrated a small instance of our network. But, in the weighted variation (WSRT), there are weights between reviews and terms according to the number of words on that review. We modeled the network in this manner, as we believe that there is a relation between helpfulness and the number of stars of a review. Thus, we also believe that there is a relation between terms, reviews, and stars.
- Network Type Review-Term (RT): In this model, there are no `stars` nodes. The `review` and `term` nodes are linked to each other whenever the term is present in the review. There are no auto-loops. Similarly to the USRT variation, the network is undirected and unweighted (URT). And there is a weighted variation (WRT) where there are weights between revisions and terms according to the number of words in that revision. In Figure 5, we illustrated a small instance of our network. We modeled this variation to evaluate if only `review` and `term` nodes are sufficient to describe the problem satisfactorily.

### 3.5. Regularization

The network generation considers the similarity information among elements of the network, for example, using a Bag-of-Words (BoW) representation and calculating the similarity of documents by the cosine similarity metric. However, it is not always possible to create a network this way. Moreover, modeling a network in this manner is unnatural.

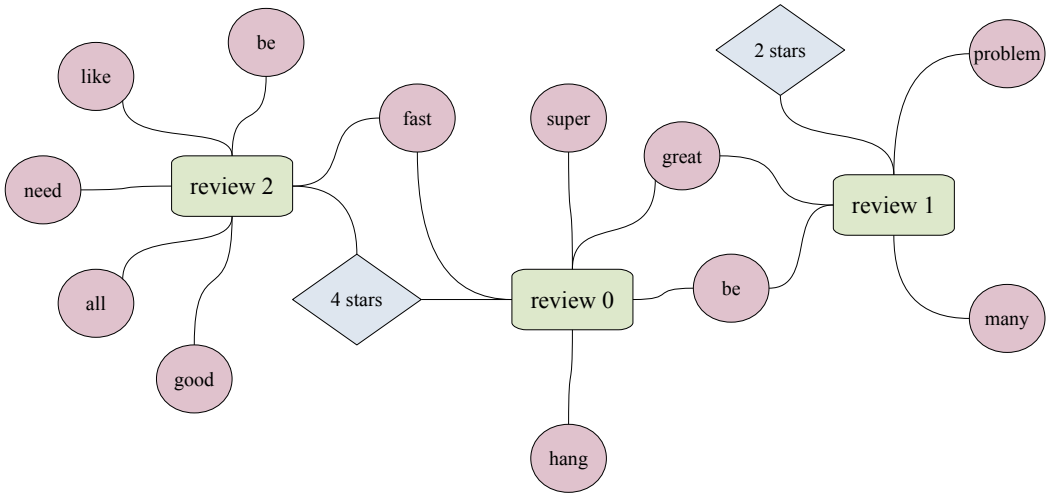



Figure 4. Network SRT instance example

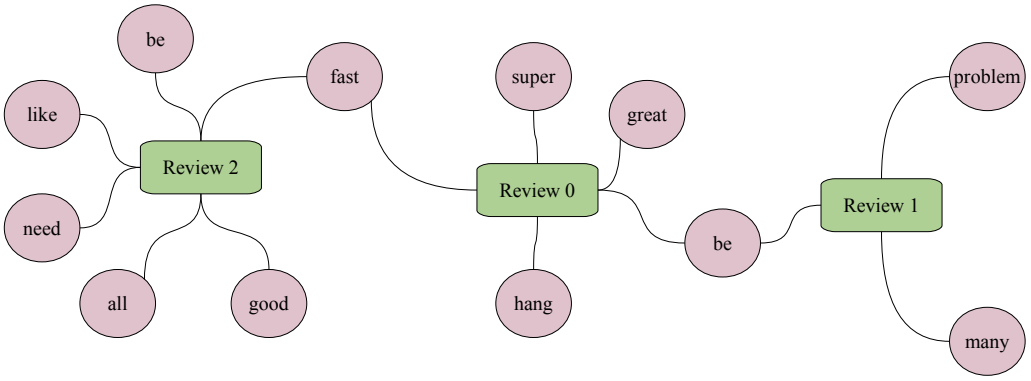



Figure 5. Network RT instance example

More than that, it is difficult to consider and aggregate domain information in the similarity measure between nodes. Hence, in these cases, the methods do not achieve good results.

In this paper, we aim to show an example where that difficulty may be overcome. In previous experiments, the similarity between document vectors did not achieve good results. Thus, we created an heterogeneous network, considering the relationship information among existing elements in the reviews. Using different type nodes increases network informativeness, but it difficulties the use of some learning methods direct in the network.

In this context, we applied regularization methods to perform the features extraction about network objects classes. Regularization is a kind of transductive classification method [Rossi 2016]. It aims to find a set of labels that follows two conditions: (i) the method needs to be consistent with the set of labels manually annotated and; (ii) it needs to be consistent with the network topology, i.e., to consider that nearest neighbors tend to have the same labels.

We adopted three methods for regularization: Gaussian Fields and Harmonic Function (GFHF) [Zhu et al. 2003], Learning with Local and Global Consistence (LLGC) [Zhou et al. 2004], and GnetMine [Ji et al. 2010]. These three methods have slight differences in their equations. For example, GFHF seeks to minimize the function in the Equation 3. For assigning a label into an object, the method computes the weighted average of its neighbors' label information by the weights of the links between objects, as presented in Equation 4, with the following terms:

- $O$ is the set of nodes in the network.
- $O^L$ is the set of pre-annotated nodes in the network.
- $F$ e $f$ is the regularization output. They represent a vector with the relative coordinates of a review in the plane (see Table 3).
- $w$ is the edge weight between nodes $o_i$ and $o_j$.
- $y$ is the information vector for the pre-annotated nodes.

$$Q(F) = \frac{1}{2} \sum_{o_i, o_j \epsilon\, O} w_{o_i,o_j}(f_{o_i} - f_{o_j})^2 + \lim_{\mu \to \infty} \mu \sum_{o_i \epsilon O^L} (f_{o_i} - y_{o_i})^2 \tag{3}$$

$$f_{o_i} = \frac{\sum_{o_j \epsilon O} w_{o_i,o_j} f_{o_j}}{\sum_{o_j \epsilon O} w_{o_i,o_j}} \tag{4}$$

The regularization algorithms requires some nodes to be manually pre-labeled with specific classes. The main difference between GFHF and LLGC is that GFHF does not modify these pre-labeled nodes, unlike the LLGC that performs such modifications. (in Section 4, we detail the labeling process). The GnetMine, in addition to modify the values of pre-labeled nodes, it is an algorithm for heterogeneous networks, so it considers the different types of nodes. The regularization algorithms produce values related to coordinates for each object in the network, as shown in Table 3. These values may be used for several supervised machine learning methods to learn and predict labels [Bui et al. 2018]. In subsection 3.6, we detail the used machine learning algorithms.

Table 3. Example of the regularization algorithm output

| Id | Coordinate 1 | Coordinate 2 |
|---|---|---|
| 0 | 0.13884248 | 0.11291029 |
| 1 | 0.13011554 | 0.12082376 |
| 2 | 0.12334355 | 0.13454545 |
| 3 | 0.12324345 | 0.12324455 |
| ... | ... | ... |

### 3.6. Classification

In this step, we classified each review as helpful or not helpful from the extracted features by regularization algorithms. That is, from the generated coordinates for each review for helpful and not helpful label, we handle the problem of helpfulness prediction as a supervised machine learning problem.

We evaluate several classifies available by the Scikit-Learn library [Pedregosa et al. 2011], such as Support Vector Machine (SVM), Naïve Bayes, C4.5, and Multi-layer Perceptron (MLP).

In what follows, we detail the performed experiments and obtained results. Besides, we compare our approach with a well-known baseline.

## 4. Experiments and Results

We perform an experiment to evaluate the impact of several possible settings of our approach. The variables of the experiment are: amount of pre-annotate nodes to regularization, type of regularization algorithm, and type of the modeling network.

To start the experiment, first, we balanced the data. For that, we applied a downsampling method that randomly assigns to the corpus the same number of helpful and not-helpful reviews, removing samples of the majority class. Next, with the balanced corpus, we explored the regularization methods, aiming to get the features of the corpus. Finally, we used four supervised classifiers with the extracted features to identify if an opinion is helpful or not. Furthermore, we compared our approach against two methods. The first one adopts an approach based on fuzzy logic [de Sousa et al. 2015], while the second one is based on NN [Santos et al. 2016]. This second method is the evolution of first one. Both methods classify reviews into four classes: insufficient, sufficient, good, and excellent. We adapted the baseline methods to consider only two classes: excellent and good as helpful, and insufficient and sufficient as not helpful. This modification was necessary since our method use only two classes: helpful and not helpful.

We adopted the works of [de Sousa et al. 2015] and [Santos et al. 2016] for comparison because they are open source and requires only minor modifications for re-use, and are domain-independent. The main difference between our current approach and previous approaches is that the latter do not consider the relationships among the opinion elements. Moreover, the previous approaches did not focus on binary opinion helpfulness classification but on multi-class classification.

In addition to the works of [de Sousa et al. 2015, Santos et al. 2016], we developed two other baseline methods to compare with our approach. In the first one (Baseline 1), we used the Bag-of-Words as features, while in the second (Baseline 2), we used three well-known features: average sentence length, number of tokens, and number of sentences. We applied both baselines into the Naïve Bayes classifier.

Since the regularization algorithms we used requires a portion of pre-annotated reviews, we followed a strategy for progressively annotate the reviews. For example, we manually annotated from 0.5% to 5.0% of the reviews as helpful and not helpful. We randomly chose these proportions. In this way, we analyzed several supervised algorithms, such as SVM, Naïve Bayes, C4.5, and MLP. To evaluate these algorithms, we applied the $k$-fold cross-validation technique with $k = 10$. Tables 4, 5, 6, and 7 show the results for each pre-annotated portion of reviews and each algorithm. We repeat these steps for each regularization algorithm.

From tables results, we can see that the MLP algorithm reached the better results

Table 4. Results for the classification algorithms for network type Unweighted SRT

| Pre-labeled comment (%) | GFHF | | | | LLGC | | | | GNETMINE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** |
| **0.5** | 0.5017 | 0.5130 | 0.5352 | 0.5500 | 0.5023 | 0.7318 | 0.6593 | 0.5005 | 0.5019 | 0.5089 | 0.5315 | 0.5023 |
| **1.0** | 0.5037 | 0.5087 | 0.5444 | 0.5362 | 0.5049 | 0.7405 | 0.6605 | 0.5003 | 0.5033 | 0.5163 | 0.5403 | 0.5047 |
| **1.5** | 0.5071 | 0.5077 | 0.5530 | 0.5579 | 0.5073 | 0.7462 | 0.6616 | 0.4995 | 0.5059 | 0.5269 | 0.5471 | 0.5070 |
| **2.5** | 0.5090 | 0.5129 | 0.5520 | 0.5831 | 0.5099 | 0.7503 | 0.6715 | 0.4986 | 0.5090 | 0.6162 | 0.5745 | 0.5099 |
| **3.0** | 0.5105 | 0.5139 | 0.5639 | 0.6265 | 0.5123 | 0.7493 | 0.6663 | 0.4997 | 0.5109 | 0.5456 | 0.5963 | 0.5121 |
| **3.5** | 0.5127 | 0.5177 | 0.5708 | 0.6159 | 0.5148 | 0.7515 | 0.6642 | 0.5013 | 0.5134 | 0.5271 | 0.6162 | 0.5147 |
| **4.0** | 0.5187 | 0.5195 | 0.5612 | 0.6238 | 0.5173 | 0.7582 | 0.6767 | 0.7646 | 0.5155 | 0.5525 | 0.6182 | 0.5176 |
| **4.5** | 0.5328 | 0.5221 | 0.5810 | 0.6351 | 0.5198 | 0.7591 | 0.6774 | 0.7664 | 0.5179 | 0.6238 | 0.6525 | 0.5273 |
| **5.0** | 0.5468 | 0.5250 | 0.5841 | 0.6253 | 0.5024 | 0.7602 | 0.6679 | **0.7723** | 0.5204 | 0.5524 | 0.6403 | 0.5245 |

Table 5. Results for the classification algorithms for network type Weighted SRT

| Pre-labeled comment (%) | GFHF | | | | LLGC | | | | GNETMINE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** |
| **0.5** | 0.5015 | 0.5178 | 0.5329 | 0.5061 | 0.5023 | 0.7379 | 0.6539 | 0.5001 | 0.5018 | 0.4843 | 0.5109 | 0.5023 |
| **1.0** | 0.5034 | 0.5114 | 0.5383 | 0.5306 | 0.5049 | 0.7396 | 0.6589 | 0.5010 | 0.5037 | 0.5868 | 0.5392 | 0.5048 |
| **1.5** | 0.5059 | 0.5118 | 0.5499 | 0.5574 | 0.5072 | 0.7474 | 0.6525 | 0.4997 | 0.5059 | 0.5346 | 0.5486 | 0.5105 |
| **2.0** | 0.5078 | 0.5130 | 0.5496 | 0.5929 | 0.5099 | 0.7500 | 0.6602 | 0.5000 | 0.5078 | 0.5276 | 0.5874 | 0.5221 |
| **2.5** | 0.5123 | 0.5153 | 0.5589 | 0.5982 | 0.5123 | 0.7522 | 0.6711 | 0.5010 | 0.5109 | 0.5742 | 0.6153 | 0.5223 |
| **3.0** | 0.5126 | 0.5164 | 0.5539 | 0.6001 | 0.5149 | 0.7585 | 0.6632 | 0.6046 | 0.5122 | 0.5724 | 0.6089 | 0.5149 |
| **3.5** | 0.5156 | 0.5173 | 0.5564 | 0.6102 | 0.5174 | 0.7552 | 0.6683 | 0.7628 | 0.5260 | 0.5703 | 0.6366 | 0.5183 |
| **4.0** | 0.5348 | 0.5211 | 0.5639 | 0.6323 | 0.5198 | 0.7548 | 0.6663 | 0.7592 | 0.5179 | 0.5786 | 0.6274 | 0.5301 |
| **4.5** | 0.5188 | 0.5237 | 0.5582 | 0.6091 | 0.5224 | 0.7587 | 0.6744 | 0.7651 | 0.5209 | 0.5857 | 0.6315 | 0.5354 |
| **5.0** | 0.5381 | 0.5258 | 0.5594 | 0.6251 | 0.5248 | 0.7631 | 0.6683 | **0.7647** | 0.5220 | 0.5628 | 0.6396 | 0.5367 |

Table 6. Results for the classification algorithms for network type Unweighted RT

| Pre-labeled comment (%) | GFHF | | | | LLGC | | | | GNETMINE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** |
| **0.5** | 0.5017 | 0.5110 | 0.5355 | 0.5147 | 0.5023 | 0.7318 | 0.6595 | 0.5004 | 0.5016 | 0.5423 | 0.5222 | 0.5023 |
| **1.0** | 0.5036 | 0.5109 | 0.5483 | 0.5498 | 0.5049 | 0.7411 | 0.6604 | 0.4998 | 0.5039 | 0.5813 | 0.5378 | 0.5046 |
| **1.5** | 0.5055 | 0.5147 | 0.5524 | 0.5461 | 0.5073 | 0.7285 | 0.6557 | 0.5002 | 0.5073 | 0.5866 | 0.5485 | 0.5073 |
| **2.0** | 0.5165 | 0.5179 | 0.5540 | 0.5699 | 0.5008 | 0.7138 | 0.6625 | 0.4989 | 0.5083 | 0.5883 | 0.5607 | 0.5092 |
| **2.5** | 0.5228 | 0.5183 | 0.5662 | 0.5766 | 0.5122 | 0.7180 | 0.6665 | 0.5012 | 0.5109 | 0.6144 | 0.5847 | 0.5153 |
| **3.0** | 0.5177 | 0.5213 | 0.5415 | 0.5822 | 0.5149 | 0.7323 | 0.6730 | 0.5517 | 0.5153 | 0.5837 | 0.6309 | 0.5153 |
| **3.5** | 0.5194 | 0.5213 | 0.5635 | 0.5833 | 0.5174 | 0.7452 | 0.6668 | 0.7632 | 0.5154 | 0.6089 | 0.6246 | 0.5241 |
| **4.0** | 0.5195 | 0.5234 | 0.5565 | 0.5779 | 0.5197 | 0.7370 | 0.6771 | 0.7645 | 0.5174 | 0.6061 | 0.6337 | 0.5313 |
| **4.5** | 0.5254 | 0.5259 | 0.5673 | 0.6119 | 0.5224 | 0.7501 | 0.6705 | 0.7673 | 0.5211 | 0.6149 | 0.6518 | 0.5226 |
| **5.0** | 0.5345 | 0.5293 | 0.5688 | 0.6173 | 0.5247 | 0.7466 | 0.6809 | **0.7685** | 0.5227 | 0.6028 | 0.6366 | 0.5271 |

Table 7. Results for the classification algorithms for network type Weighted RT

| Pre-labeled comment (%) | GFHF | | | | LLGC | | | | GNETMINE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** | **SVM** | **Naïve Bayes** | **C4.5** | **MLP** |
| **0.5** | 0.5010 | 0.5073 | 0.5395 | 0.5132 | 0.5023 | 0.7308 | 0.6600 | 0.5011 | 0.5014 | 0.5331 | 0.5336 | 0.5023 |
| **1.0** | 0.5034 | 0.5117 | 0.5432 | 0.5449 | 0.5048 | 0.6869 | 0.6532 | 0.5022 | 0.5039 | 0.5756 | 0.5399 | 0.5049 |
| **1.5** | 0.5073 | 0.5139 | 0.5451 | 0.5331 | 0.5073 | 0.7461 | 0.6619 | 0.5000 | 0.5057 | 0.5780 | 0.5570 | 0.5078 |
| **2.0** | 0.5082 | 0.5209 | 0.5526 | 0.5406 | 0.5099 | 0.7423 | 0.6604 | 0.5016 | 0.5086 | 0.5836 | 0.5636 | 0.5115 |
| **2.5** | 0.5286 | 0.5210 | 0.5521 | 0.5932 | 0.5122 | 0.7249 | 0.6708 | 0.5008 | 0.5109 | 0.5941 | 0.6089 | 0.5128 |
| **3.0** | 0.5176 | 0.5196 | 0.5574 | 0.5633 | 0.5149 | 0.7540 | 0.6661 | 0.7575 | 0.5125 | 0.5871 | 0.6137 | 0.5230 |
| **3.5** | 0.5205 | 0.5227 | 0.5631 | 0.5951 | 0.5173 | 0.7359 | 0.6737 | 0.7599 | 0.5139 | 0.5988 | 0.6441 | 0.5336 |
| **4.0** | 0.5237 | 0.5253 | 0.5734 | 0.6065 | 0.5196 | 0.7359 | 0.6749 | 0.7636 | 0.5179 | 0.6130 | 0.6381 | 0.5319 |
| **4.5** | 0.5260 | 0.5276 | 0.5783 | 0.6053 | 0.5224 | 0.7578 | 0.6795 | 0.7634 | 0.5208 | 0.6142 | 0.6368 | 0.5234 |
| **5.0** | 0.5285 | 0.5285 | 0.5689 | 0.6025 | 0.5246 | 0.7502 | 0.6755 | **0.7652** | 0.5221 | 0.6053 | 0.6422 | 0.5316 |

in all regularization algorithms and variations of networks, achieving the best result with 5.0% of pre-annotated reviews. The LLGC regularization achieved the best result in all

networks variations. It is important to say that we performed experiments with more pre-annotated reviews, but this did not improve the results, the tables results show that fact, the improving of results after 4.0% of pre-annotated reviews are low. An important observation is that all variations of networks achieved close results, the difference is really small, mainly for the best ones.

We also carried out experimentation using a holdout approach, splitting the corpus in 80% to train and 20% to test. In Table 8, we present the results of the MLP for each class, showing precision, recall, and f-measure. One may see that the MLP produced closely results for each class.

Finally, we compared our approach with the developed baselines, as shown in Table 9. From this table, we compared our network with MLP against the baselines. We can see that our method outperformed the best baseline in 0.5 of f-score.

Table 8. Results of the MLP for LLGC regularization algorithm and USRT network type (Holdout)

| Label | Precision | Recall | F1 |
|---|---|---|---|
| Helpful | 0.81 | 0.62 | 0.70 |
| Not helpful | 0.69 | 0.85 | 0.76 |

Table 9. Comparison among approaches (cross-validation)

| Approach | F1 |
|---|---|
| Baseline 1 | 0.72 |
| Baseline 2 | 0.63 |
| Fuzzy | 0.60 |
| RNA | 0.58 |
| Our network | **0.77** |

These evaluations show the power of the network-based approach to model and predict the helpfulness of opinions. However, it is necessary to investigate other topologies, specific metrics, and adapt other features used in other languages.

Also it is important to highlight that we performed a test of significance in order to verify if our results are statistically significant and different from the baselines. We found out a p-value $< 0.05$, indicating that the results are statistically different and significant with 95% confidence.

## 5. Conclusion and Future Work

Most papers in the literature model the opinions helpfulness prediction task as an attribute-value table. In this way, the relationship information among objects did not consider. Our hypothesis is that, it is possible to improve the results of the helpfulness prediction using relationship information among opinion elements. In this paper, we modeled the opinions helpfulness prediction task as a heterogeneous network. From this network, we applied three regularization algorithms for feature extraction from user reviews. At last, we evaluated several supervised machine learning algorithms on the extracted features. We compared our approach with four baselines methods. For that, we performed an experiment on $100,332$ reviews about apps from Google Play belong to the UTLCorpus. The results showed that our network outperformed the baselines.

Despite the good results, it is important to say that our approach is network-modeling dependent, however, to use neural-graph machines allow to classifier a larger

dataset with few annotated data. The close results among the regularization methods, indicating that it is possible to adopt any regularization method for the classification.

The main contribution of this work is the modeling of a heterogeneous network for the opinions helpfulness prediction task. The informativity power of a heterogeneous network is a great differential in relation to based attribute-value approaches. Although our approach is relatively new, the obtained results showed their benefits and potentialities. However, as this work is a pioneer in considering the helpfulness task for Portuguese, a comparison with other works becomes difficult. However, as future work, we intend to adapt methods from other languages to Portuguese, aiming to compare them with our approach.

For future work, besides to apply the approach on multi-domains, we intend to follow two research lines. In the first, we will explore linguistically motivated approaches, that is, we will evaluate linguistic features that indicate if a review is helpful or not. In the second, we will adapt works of other languages, aiming to investigate language independent-features.

In addition to these two directions, we will explore other network topologies in order to achieve better results. For example, topologies with many layers (a common layer and a layer with semantic information: synonyms, named entities, sentiment words, etc.). Furthermore, these topologies may allow the use of several network metrics, such as hubs, betweenness, and closeness, among others. And, finally, to explore the use of Deep Neural Networks applied to regularization extracted features.

## 6. Acknowledges

The authors are grateful to the IFPI for supporting this work.

## References

Anchiêta, R., Sousa, R. F., Moura, R., and Pardo, T. (2017). Improving opinion summarization by assessing sentence importance in on-line reviews. In *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*, pages 32–36.

Anchiêta, R. T. and Moura, R. S. (2017). Exploring unsupervised learning towards extractive summarization of user reviews. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, pages 217–220. ACM.

Barbosa, J. L. and Moura, R. S. (2016). Avaliaç ao automática da utilidade de reviews usando redes neurais artificiais no corpus do steam. In *Anais do XXVI Congresso da Sociedade Brasileira de Computação: BraSNAM - 5º Brazilian Workshop on Social Network Analysis and Mining*. Brazilian Computer Society.

Bertaglia, T. F. C. and Nunes, M. d. G. V. (2016). Exploring word embeddings for unsupervised textual user-generated content normalization. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 112–120.

Bui, T. D., Ravi, S., and Ramavajjala, V. (2018). Neural graph learning: Training neural networks using graphs. In *Proceedings of 11th ACM International Conference on Web Search and Data Mining (WSDM)*.

de Sousa, R., Anchieta, R., and Nunes, M. (2019). Um método baseado em grafos para predição da utilidade de opiniões sobre produtos. In *Anais do VIII Brazilian Workshop on Social Network Analysis and Mining*, pages 95–106, Porto Alegre, RS, Brasil. SBC.

de Sousa, R. F., Rabêlo, R. A., and Moura, R. S. (2015). A fuzzy system-based approach to estimate the importance of online customer reviews. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*, pages 1–8. IEEE.

Diaz, G. O. and Ng, V. (2018). Modeling and prediction of online product review helpfulness: A survey. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 698–708.

Fonseca, E. R. and Rosa, J. L. G. (2013). Mac-morpho revisited: Towards robust part-of-speech tagging. In *Proceedings of the 9th Brazilian symposium in information and human language technology*, pages 98–107.

Hartmann, N. S., Avanço, L. V., Balage Filho, P. P., Duran, M. S., Nunes, M. D. G. V., Pardo, T. A. S., Aluisio, S. M., et al. (2014). A large corpus of product reviews in portuguese: Tackling out-of-vocabulary words. In *International Conference on Language Resources and Evaluation*. European Language Resources Association-ELRA.

Ji, M., Sun, Y., Danilevsky, M., Han, J., and Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer.

Kim, S.-M., Pantel, P., Chklovski, T., and Pennacchiotti, M. (2006). Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on empirical methods in natural language processing*, pages 423–430. Association for Computational Linguistics.

Krishnamoorthy, S. (2015). Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, 42(7):3751–3759.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Liu, J., Cao, Y., Lin, C.-Y., Huang, Y., and Zhou, M. (2007). Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Malik, M. and Hussain, A. (2017). Helpfulness of product reviews as a function of discrete positive and negative emotions. *Computers in Human Behavior*, 73:290–302.

Martins, A. C. S. and Tacla, C. A. (2015). Assessement of features influencing the voting for opinions' helpfulness about services in portuguese. In *Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective-Volume 1*, page 21. Brazilian Computer Society.

Orengo, V. and Huyck, C. (2001). A stemming algorithmm for the portuguese language. In *String Processing and Information Retrieval*, pages 186–193.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rossi, R. G. (2016). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis, Universidade de São Paulo.

Santos, R. L. d. S., de Sousa, R. F., Rabelo, R. A., and Moura, R. S. (2016). An experimental study based on fuzzy systems and artificial neural networks to estimate the importance of reviews about product and services. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 647–653. IEEE.

Scarton, C. E. and Aluísio, S. M. (2010). Análise da inteligibilidade de textos via ferramentas de processamento de língua natural: adaptando as métricas do coh-metrix para o português. *Linguamática*, 2(1):45–61.

Semin, G. R. (2011). The linguistic category model. *Handbook of theories of social psychology*, 1:309–326.

Singh, J. P., Irani, S., Rana, N. P., Dwivedi, Y. K., Saumya, S., and Roy, P. K. (2017). Predicting the "helpfulness" of online consumer reviews. *Journal of Business Research*, 70:346–355.

Sousa, R. F., Brum, H. B., and Nunes, M. d. G. V. (2019). A bunch of helpfulness and sentiment corpora in brazilian portuguese. In *Proceedings of the 12th Brazilian Symposium in Information and Human Language Technology*, pages 209–218. Sociedade Brasileira de Computação.

Zeng, Y.-C., Ku, T., Wu, S.-H., Chen, L.-P., and Chen, G.-D. (2014). Modeling the helpful opinion mining of online consumer reviews as a classification problem. *International Journal of Computational Linguistics & Chinese Language Processing, Volume 19, Number 2, June 2014*, 19(2).

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.