UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

VINICIUS WOLOSZYN

Unsupervised Learning Strategies for Automatic Generation of Personalized Summaries

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dr. Leandro Krug Wives

Porto Alegre
May 2019

CIP — CATALOGING-IN-PUBLICATION

Woloszyn, Vinicius

Unsupervised Learning Strategies for Automatic Generation of Personalized Summaries / Vinicius Woloszyn. – Porto Alegre: PPGC da UFRGS, 2019.

78 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2019. Advisor: Leandro Krug Wives.

1. Unsupervised learning. 2. Text summarization. 3. Personalization. 4. Bias. I. Wives, Leandro Krug. II. Título.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Dr. Leandro Krug Wives for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank my family and my parents for supporting me spiritually throughout writing this thesis and my my life in general.

ABSTRACT

It is relatively hard for readers to deal objectively with large documents in order to absorb the key idea about a particular subject. In this sense, automatic text summarization plays an important role by systematically digest a large number of documents to produce in-depth abstracts. Despite fifty years of studies in automatic summarization of texts, one of the still persistent shortcomings is that the individual interests of the readers are still not considered. Regarding the automatic techniques for generation of summaries, it mostly relies on supervised Machine Learning algorithms such as classification and regression, however, the quality of results is dependent on the existence of a large, domain-dependent training data set. On the other hand, unsupervised learning strategies are an attractive alternative to avoid the labor-intense and error-prone task of manual annotation of training data sets. To accomplish such objective, this work puts forward a novel unsupervised and semi-supervised algorithms to automatically generate tailored summaries. Our experiments showed that we can effectively identify a significant number of interesting passages for the readers with less data for the training step.

Keywords: Unsupervised learning. text summarization. personalization. bias.

Métodos não-supervisionados para a geração Automática de Sumários Personalizados

RESUMO

É relativamente difícil para leitores lidarem objetivamente com grandes documentos para absorver a ideia-chave sobre um determinado assunto. Nesse sentido, técnicas automáticas para sumarização de texto desempenham um papel importante ao digerir sistematicamente um grande número de documentos para produzir resumos detalhados. Apesar dos resumos gerados por máquina terem mais de cinquenta anos, uma das falhas é que geralmente seus métodos não consideram o interesse dos leitores durante o processo de criação, culminando em resumos de propósito geral. Em relação às técnicas, normalmente a sumarização automática de textos baseia-se em algoritmos de Aprendizado de Máquina supervisionados, como classificação e regressão. No entanto, a qualidade dos resultados depende da existência de um grande conjunto de dados de treinamento dependente de domínio. Por outro lado, as estratégias de aprendizado não supervisionadas são uma alternativa atraente para evitar a tarefa intensa de trabalho e propensa a erros de anotação manual de conjuntos de dados de treinamento. Este trabalho realiza uma análise abrangente de algoritmos de Aprendizado de Máquina não supervisionados para gerar, automaticamente, um Resumo Personalizado.

Palavras-chave: aprendizado não supervisionado, sumarização de texto, análise de viés.

LIST OF FIGURES

Figure 1.1 Pipeline of this Thesis. 13

Figure 2.1 Illustration of MRR steps, where symbols represent text words and numbers, star ratings. 17
Figure 2.2 Distribution of results obtained in MRR and the baseline on books reviews. 24
Figure 2.3 Distribution of results obtained from MRR and the baseline on electronics reviews. 24
Figure 2.4 Graph-Specific Threshold versus different values for Fixed Thresholds. 25
Figure 2.5 Influence of MRR's parameters on NDCG results 26
Figure 2.6 Run-time comparison between MRR, REVRANK and PR_HS_LEN for electronic products reviews. 27

Figure 3.1 A summarized snapshot of "Into the Wild" lesson plan 36
Figure 3.2 Distribution of Rouge results. 39

Figure 4.1 The distribution of the URL similarity between false and true News domains, where * represent the mean. 44
Figure 4.2 Distribution of collected URLs per category of News, where the categories were extracted from http://similarweb.com/ 49
Figure 4.3 Year's distribution of collected News, ranging from 2010 to 2018. 50
Figure 4.4 Distribution of URL's number collected per domain. 51
Figure 4.5 Jaccard Similarity between News categories and fake News that achieve the minimum similarity (>0.4). 51
Figure 4.6 Number of seeds used to train the model. 53

Figure 5.1 Similarity using Jaccard Index of the user's review with a Summary, The Most Helpful Review, and All Reviews about the same product 61
Figure 5.2 A simple text graph. 63
Figure 5.3 Interest dampening. 65

LIST OF TABLES

Table 2.1 Profiling of the Amazon dataset. 20
Table 2.2 Mean Performance on Book Reviews 23
Table 2.3 Mean Performance on Electronic Reviews 23

Table 3.1 Amazon Movie Reviews Statistics 32
Table 3.2 Keywords extracted from the lesson plans in TWM 38
Table 3.3 Mean of ROUGE results achieved by BEATnIk and the Baseline....38
Table 3.4 Snippets of the summaries generated by BEATnIk and the Baseline about movie 'Conrack' 39

Table 4.1 Reliable News' URLs, their headlines and Extracted Terms extracted....47
Table 4.2 Summary of the reliable and Unreliable News Websites used in this work....50
Table 4.3 Confusion Matrix of DistrustRank. 54
Table 4.4 Confusion Matrix of SVM. 54
Table 4.5 Summary of Results 55

Table 5.1 Profiling of the Amazon dataset. 67
Table 5.2 Mean Performance using Jaccard Similarity Index, where IR means InterestRanking. 69

CONTENTS

1 INTRODUCTION 10
1.1 Research problem 10
1.2 Thesis Overview 12
2 RANKING DOCUMENTS BY RELEVANCE 14
2.1 Introduction 14
2.2 MRR Algorithm 16
2.2.1 Reviews Similarity 17
2.2.2 Graph-Specific Similarity Threshold 18
2.2.3 PageRank Centrality 19
2.3 Experiment Design 20
2.3.1 Datasets 20
2.3.2 Baseline 21
2.3.3 Gold Standard and Evaluation Metric 21
2.3.4 Normalized Discounted Cumulative Gain 21
2.3.5 MRR Parameterization 22
2.3.6 Graph-Specific Threshold Assessment 22
2.3.7 Run-time Evaluation 22
2.4 Results and Discussions 23
2.4.1 Relevance Ranking Assessment 23
2.4.2 Graph-Specific Threshold Assessment 25
2.4.3 Parameter Sensitivity Assessment 25
2.4.4 Run-time Assessment 26
2.5 Related Work 26
2.6 Chapter summary 28
2.7 Final Remarks 29
3 BIASED SUMMARIZATION 30
3.1 Introduction 30
3.2 Datasets Employed 31
3.2.1 Teaching with Movies 32
3.2.2 Amazon Movie Reviews 32
3.3 BEATnIk Algorithm 33
3.4 Experiment Design 35
3.4.1 The baseline 35
3.4.2 Gold-Standard 36
3.4.3 Evaluation Metric 37
3.4.4 BEATnIk's bias 37
3.5 Results 37
3.6 Related work 39
3.7 Chapter summary 40
3.8 Final Remarks 41
4 RANKING DOCUMENTS BY TRUST 42
4.1 Introduction 42
4.2 DistrustRank Algorithm 44
4.2.1 Similarity between Websites 45
4.2.2 Similarity Threshold (β) 46
4.2.3 Biased Centrality 47
4.3 Experiment Design 48
4.3.1 Datasets 48

4.3.2 Validation 50
4.3.3 Defining a Similarity Threshold (β) 51
4.3.4 Metrics 52
4.3.5 Baseline 52
4.4 Results and Discussion 52
4.4.1 Ranking Task Assessment 53
4.4.2 Classification Task Assessment 54
4.5 Related Work 55
4.6 Final remarks 57
4.7 Chapter summary 58
5 PERSONALIZATION OF SUMMARIES 59
5.1 Introduction 59
5.2 Use Case 60
5.3 InterestRanking Algorithm 61
5.3.1 Text model 62
5.3.2 Similarity Threshold (β) 62
5.3.3 Interests 63
5.3.4 Biased Centrality 64
5.3.5 Interest Attenuation 64
5.3.6 Diversity 65
5.3.7 Algorithm 65
5.4 Experimental Design 67
5.4.1 Data set 67
5.4.2 Gold Standard 67
5.4.3 Jaccard Similarity Index 68
5.4.4 Baseline 68
5.5 Results and Discussion 68
5.6 Related Work 69
5.7 Conclusion and Future Work 69
5.8 Threads to the validity 70
5.9 Final Remarks 70
6 CONCLUSIONS 72
REFERENCES 74

1 INTRODUCTION

Automatic Text Summarization (ATS) techniques have been widely employed in order to systematically digest a large number of documents and generate in-depth abstracts. Despite fifty years of studies in automatic summarization of texts, one of the still persistent shortcomings is that the individual interests of the readers are not considered. Furthermore, the automatic generation of personalized summaries, which meet the individual profile of the readers, is still underexplored and remains an open research problem.

Regarding techniques, many rely on supervised learning strategies such as classification and regression (XIONG; LITMAN, 2011; ZENG; WU, 2013; YANG et al., 2015; WOLOSZYN et al., 2017a). On the one hand, a common drawback of supervised learning strategies is that the quality of results is greatly influenced by the availability of a large, domain-dependent annotated corpus for the training step. On the other hand, unsupervised techniques are an attractive alternative to avoid the labor-intense and error-prone task of manual annotation of training data sets when this is not available (HSUEH; MELVILLE; SINDHWANI, 2009).

With these two problems at hand, this thesis puts forward a new possibility for machine-generated summaries: **personalization**. Naturally, for an automatic generation of a personalized abstract, information about the reader is needed. However, for the best of our knowledge, a specific gold standard suitable for this problem does not exist yet. To overcome this limitation, we propose to explore unsupervised, and semi-supervised learning strategies, since these do not rely on a manually annotated training set. Additionally, those strategies have shown a comparable performance in related tasks when compared to supervised models (WU; XU; LI, 2011; WOLOSZYN et al., 2017b). For validation purposes, we employ data collected from collaborative review websites (e.g., *Amazon.com and Goodreads.com*), since they provide a rich source of information which contains preferences, choices - i.e., what a particular user likes and dislikes. Reviews have also been widely employed in other fields, for instance, Recommender Systems for the generation of recommendations.

1.1 Research problem

As introduced before, this thesis addresses the problem of selecting the most relevant information from documents to compose a tailored summary. In this Section, we

explicitly define the research problems, as well as the hypotheses, as follows:

- **RQ 1** *How to detect a relevant document among a large number of documents?*
  Analog to this question is the problem of finding 'The Most Helpful Review'. In collaborative review websites, relevance is often measured in terms of helpfulness, which is interpreted as the perceived value of a given entry to support purchasing decisions (MUDAMBI; SCHUFF, 2010). Normally, users are invited to give their feedback about the relevance of other reviews using straightforward questions such as "Was this review helpful to you?"; the most voted one becomes 'The Most Helpful Review', and it is usually featured prominently on the website.
  Many studies have been addressed to understand the dimensions of a relevant document to then automatic predict its relevance. Commonly, the approaches in this domain rely on supervised learning strategies using the human vote as a gold standard (XIONG; LITMAN, 2011; ZENG; WU, 2013; YANG et al., 2015; KIM et al., 2006; MUKHERJEE; LIU, 2012; TANG; QIN; LIU, 2015). However, supervised learning approaches which predict the relevance of documents depends on the availability of a large, domain-dependent annotated corpus. Besides, since manual process are dependent on motivated users, often those approaches fail to consider the most recent reviews which naturally have fewer votes comparing to the old ones, configuring the so-called "cold start problem" (LAM et al., 2008).
- **RQ 2** *How to create a textual summary that covers the desirable information by a reader?*
  Users commonly have to deal with unrelated texts to find what they are looking for. For instance, in collaborative review websites, readers can examine the Most Helpful Review and still in many cases these do not contain all the relevant information a particular reader is interested in - this fact will be discussed in more detail in Section 8. Readers find themselves in a situation where extra research in needed in order to find the desired information. In this scenario, an abstract that covers not only the key points of a set of documents, but also relevant information for a particular reader would be valuable.

Taking into consideration the complexity and the different dimensions of the problems before asserted, our initial hypotheses are stated as follows:

- **Hypothesis 1** *Relevant documents have a higher graph centrality index since they are similar to many other documents.*

The intuition behind this hypothesis is that the relevance of a document can be regarded as the problem of finding documents that comprehend passages often highlighted. Our proposed algorithm - see Section 2 relies on the concept of graph centrality to rank documents according to their estimated relevance. To validate such assumption, we used reviews from different categories of products collected from *Amazon*. The relationship between reviews is represented as a graph, in which the vertices are the reviews, and the edges are defined in terms of the similarity between pairs of reviews. The similarity function is defined based on textual features that can be commonly found and extracted in other domains. The centrality index produces a ranking of vertices' importance, which in this approach indicates the ranking of the most relevant document.

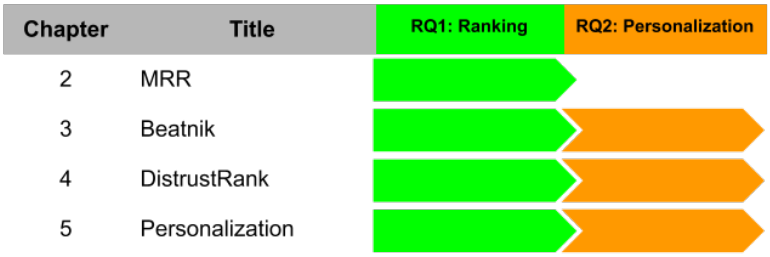
- **Hypothesis 2** *Automatic Text Summarization biased by past users' interests would generate personalized summaries covering the information sought by a reader.*
  The intuition behind this hypothesis is that we can rely on users' profiles to predict their interests and then generate personalized summaries, just like in Recommender Systems (RS) which have been used to predict users' interests based on their past behavior. Using ATS based on Graph Centrality - as stated in hypotheses 1 - with biased coverage of the user's interests could generate a short text that mimics people's textual review. The expected outcome is a short text that does not only covers the most highlighted passages of the text, but also information that the user has an interest in.

1.2 Thesis Overview

Figure 1.1 presents a broad pipeline, summarising how next Chapters are connected. Regarding Research Question 1 'How to detect a relevant document among a large number of documents?', Chapter 2 presents MMR, a novel unsupervised algorithm addressed to rank documents by relevance and address RQ1.

Regarding Research Question 2 'How to create a textual summary that covers the desirable information for a specific user?', Chapter 3 presents BEATnIk, a novel algorithm that generates biased summaries that covers user's interests. Chapter 4 describes DistrustRank, an innovative semi-supervised algorithm that identifies unreliable textual content, a plausible solution to avoid fake statements in summaries. Chapter 5 presents the proposed approach to solve the second (and last) Research Question stated here: How

Figure 1.1: Pipeline of this Thesis.



to create a textual summary of reviews that covers user's desirable information. Finally, Chapter 6 summarizes our conclusions and presents future research directions.

2 RANKING DOCUMENTS BY RELEVANCE

Identifying the most important documents is an important step towards the main goal of this thesis regarding Research Question 1 - How to detect a relevant document among a large number of documents. Thus, this Chapter put forward a novel unsupervised algorithm to rank documents by relevance. Please note that the content of this chapter was published on the IEEE/WIC/ACM International Conference on Web Intelligence[1], and here, we present an expanded version.

2.1 Introduction

Gathering information based on other people's opinions is an important part of the decision-making process. In this sense, writing and reading reviews about a service or a product on collaborative review Websites has become a common practice among people. Most collaborative review Websites adopt ranking schemes to help in dealing with information overload, using criteria such as usefulness. Usefulness, typically measured as the total votes given by other users, is an interesting way of defining the relevance of a review. However, in addition to being a manual process dependent on motivated users, this approach fails to consider the most recent reviews and those with fewer votes, configuring the so-called "cold start problem" (LAM et al., 2008).

Many works address the problem of ranking reviews by their relevance. Most of them rely on supervised algorithms such as classification and regression (KIM et al., 2006; XIONG; LITMAN, 2011; MUKHERJEE; LIU, 2012; ZENG; WU, 2013; YANG et al., 2015; TANG; QIN; LIU, 2015; CHUA; BANERJEE, 2016). However, the quality of results produced by supervised algorithms is dependent on the existence of a large, domain-dependent training data set. In this sense, unsupervised methods (TSUR; RAPPOPORT, 2009; WU; XU; LI, 2011) are an attractive alternative to avoid the labor-intense and error-prone task of manual annotation of training datasets.

In this sense, MRR (Most Relevant Reviews), a novel unsupervised algorithm that identifies relevant reviews based on the concept of node *centrality* is proposed. In graph theory, centrality (or salience) indicates the relative importance of one vertice in relation

[1]Complete Reference: Woloszyn, V., dos Santos, H. D., Wives, L. K., Becker, K. MRR: an unsupervised algorithm to rank reviews by relevance. In: Proceedings... International Conference on Web Intelligence, ACM. 2017. pp. 877-883.

to other vertices (WEST et al., 2001). Popular algorithms to calculate node centrality are PageRank (PAGE et al., 1999), and HITS (KLEINBERG, 1999).

In MRR, centrality is defined in terms of textual and rating similarity among reviews. The intuition behind this approach is that central reviews highlight aspects of a product that many other reviews frequently mention, with similar opinions, as expressed in terms of ratings. Central reviews are thus relevant because they act as a summary of a set of reviews. MRR constructs a graph where reviews are represented by nodes, connected by edges weighted by the similarity between the pair of reviews, and then employs PageRank to compute the centrality. MRR takes into account domain differences, by defining a minimum similarity threshold based on the characteristics of a set of reviews (e.g. books, movies).

Related works have explored centrality to analyze reviews based on the similarity of sentences that compose a set of reviews. For instance, RevRank (TSUR; RAPPOPORT, 2009) uses Virtual Core Review that uses centrality to rank relevant reviews by their relevance. To rank the relevance of reviews, the unsupervised approach proposed in (WU; XU; LI, 2011) combines the centrality scores assigned to individual sentences and the review's length to produce an overall centrality score for each review. The method does not scale well due to the chosen centrality granularity, which implies double use of PageRank, and required pre-processing to identify specific textual features (e.g. nouns, adjectives).

In this proposal, experiments were carried out using reviews collected from Amazon's website in two domains, and they reveal that MRR significantly outperforms the chosen unsupervised baselines (WU; XU; LI, 2011; TSUR; RAPPOPORT, 2009), both in terms of mimicking the human user perception of helpfulness and run-time performance. Comparing to a supervised baseline (Support Vector Machine regression), it achieved comparable results in a specific setting (i.e. best-ranked review).

The contributions of this work are the following:

1. an unsupervised method to identify the relevance of reviews, i.e. it does not depend on an annotated training set;
2. the use of centrality scores that rely on a computationally inexpensive similarity function that combines similarity scores of reviews, which does not require extensive textual pre-processing;
3. a method that performs well in reviews of different domains (e.g. close vs. open-ended), as it defines a graph-specific minimum similarity threshold to construct the reviews graph;

4. the use of reviews from two distinct domains, showing that MRR results are significantly superior to the unsupervised baselines, and comparable to one supervised approach in a specific setting.

The next Section 5.6 discusses related work. Then, section 3.3 present details of MRR algorithm. Section 5.4 describes the design of the experiments, and Section 5.5 discusses the results. Section 5.7 summarizes the findings up to the moment and presents future research directions.

2.2 MRR Algorithm

The intuition behind MRR[2] is that the relevance of a review can be regarded as the problem of finding reviews that comment on aspects often highlighted about that product/service, such that their rating scores do not differ much from a consensus on such aspects. To solve this problem, the MRR approach relies on the concept of graph centrality to rank reviews according to estimated relevance. Since the approach addresses the cold start problem, it does not employ features that depend on the user's indication of the received usefulness of the review (e.g. votes and author's relevance).

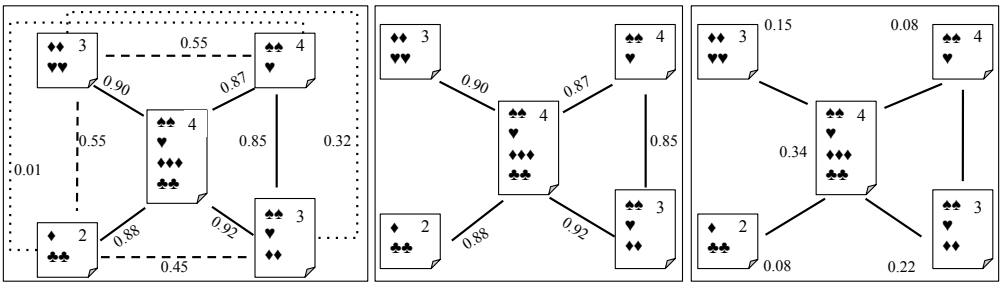
MRR represents the relationship between reviews as a graph, in which the vertices are the reviews, and the edges are defined in terms of the similarity between pairs of reviews. A similarity function that combines the similarity of topics discussed in the texts of the reviews, and the similarity of the respective rating scores, is defined. The hypothesis is that a relevant review has a high centrality index since it is similar to many other reviews. The centrality index produces a ranking of vertices' importance, which in the proposed approach indicates the ranking of the most relevant reviews.

Let R be a set of reviews, and $r \in R$ a tuple $\langle t, rs \rangle$, where $r.t$ represents the text of the review and $r.rs$ a rating score $\in [1,5]$ that the reviewer has assigned to it. MRR builds a graph representation $G = (V,E)$, where $V = R$ and E is a set of edges that connects pairs $\langle u,v \rangle$ where $v,u \in V$, and uses PageRank to calculate centrality scores for each vertex. Figure 2.1 shows the main steps of the MRR algorithm: (a) it builds a similarity graph G between pairs of reviews of the same product; (b) the graph is pruned (G') by removing all edges that do not meet a minimum similarity threshold, which is calculated based on the average similarity between reviews in the dataset; (c) using PageRank, the

[2]MRR is available at http://github.com/vwoloszyn/MRR

centrality scores are calculated and used to construct a ranking. The pseudo-code of MRR is displayed in Algorithm 4, where G and G' are represented as adjacency matrices W and W'. In the remaining of this section, the similarity function, and the process to obtain the centrality index ranking are detailed.

Figure 2.1: Illustration of MRR steps, where symbols represent text words and numbers, star ratings.



(A) Similarity Function (B) Graph-Specific Threshold (C) PageRank Scores

Algorithm 1 - MRR Algorithm (R, α, β): S

- Input: a set of reviews R, α the balance for the weighted sum in the similarity function and, β is the base threshold.
- Output: ordered list S containing the computed helpfulness score relative to each of review $\in R$.

for each $u, v \in R$ **do**
    $W[u,v] \leftarrow \alpha * sim\_txt(u,v)$+(1-$\alpha$)$* sim\_star(u,v)$
end for
$\overline{E} \leftarrow mean(W)$
for each $u, v \in R$ **do**
    if $W[u,v] \geq \overline{E} * \beta$ **then**
        $W'[u,v] \leftarrow 1$
    else
        $W'[u,v] \leftarrow 0$
    end if
end for
$S \leftarrow PageRank(W')$
Return S

2.2.1 Reviews Similarity

The premise underlying the centrality concept is that the importance of a node is measured in terms of both the number and the importance of their neighbor (which in this case, are the similar reviews). In MRR, to compute the similarity of pairs of reviews, it takes into consideration their text, disregarding its division into sentences, and the rating

scores. In addition, MRR compares the text of reviews merely using the terms they contain, represented as unigrams weighted by Term *Frequency-Inverse Document Frequency* (TF-IDF). This choice of a minimalist model, which needs only two features to represent the similarity between reviews, proved to be fast and scalable, since the extraction of features for comparison is not time-consuming. Additionally, this model achieves better results than the others two unsupervised baselines that also are based on graph centrality.

Therefore, the similarity of reviews is defined as the weighted sum between texts similarity (given by the cosine similarity of their respective TF-IDF vectors) and the similarity of ratings, as detailed in Equation 4.1.

$$f(u,v) = \alpha * sim\_txt(u,v) + (1-\alpha) * sim\_star(u,v) \tag{2.1}$$

where $sim\_txt \in [0,1]$ represents the cosine similarity between the *TF-IDF* vectors of two reviews u and v, and $sim\_star \in [0,1]$ represents the similarity between the rating scores $u.rs$ and $v.rs$. Function *sim_star*, stated in 2.2, is based on the euclidean distance normalized by the Min-Max scaling, which outputs 1 when the ratings scores $u.rs$ and $v.rs$ are identical, and 0 when ratings scores are strongly dissimilar. The constant α balances the weighted sum function. In section 2.3.5, the numerical optimization process employed to find the best α which minimizes the Mean Square Error is discussed.

$$sim\_star(u,v) = 1 - \frac{|u.rs - v.rs| - min(rs)}{max(rs) - min(rs)} \tag{2.2}$$

2.2.2 Graph-Specific Similarity Threshold

In the proposed approach, relevance is dependent on the existence of links between reviews, and it can establish links when the similarity score is above a minimum similarity threshold.

However, setting an appropriate similarity threshold is a tricky problem. While a low threshold may mistakenly consider as similar reviews that have very little in common for the computation of reviews relevance, conversely, a high threshold may disregard important links between reviews. Indeed, there may exist a significant difference on how reviews are written depending on the domain. For instance, reviews on books or movies are more diverse and open-ended, whereas reviews on computers or cameras tend to evaluate specific aspects. Thus, reviews can be more or less similar depending on the domain.

To solve this issue, a specific threshold is computed to each graph of a product, based on the characteristics of the respective reviews set. Then, a base threshold β is employed, and it varies according to the mean similarity in each reviews dataset. This base represents a proportion of the mean similarity in the dataset corresponding graph. Thus, when the mean similarity between reviews increases (decreases), the similarity threshold also increases (decreases) proportionally. The constant α balances the weighted sum function.

Equation 5.1 is used to prune the graph based on a minimum similarity between reviews. The result is an un-weighted graph represented by the adjacency matrix W', where $W'(u,v)$ assumes 1 if an edge that connects u and v exists, and 0 otherwise.

$$W'(u,v) = \begin{cases} 1, & f(u,v) \geq \overline{E} * \beta \\ 0, & otherwise \end{cases} \tag{2.3}$$

In Equation 5.1, $f(u,v)$ is the similarity score according to Equation 4.1; $\overline{E}$ is the mean similarity of the review dataset, and β is the base threshold. Section 2.3.5 present the numerical optimization process to find the best β.

2.2.3 PageRank Centrality

To compute the centrality of each review, MRR relies on PageRank (PAGE et al., 1999), which considers each edge as a vote to determine the overall centrality score of each node in a graph. However, as in many types of social networks, not all of relationships are considered of equal importance. The premise underlying PageRank is that the importance of a node is measured in terms of both the number and the importance of vertices it is related to. The PageRank centrality function is given by:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{N_v} \tag{2.4}$$

where B_u is the set containing all neighborhood of u and N_v the number of neighborhoods of v.

The intuition of using PageRank in MRR is that the more a review is connected to reviews that are highly similar to other reviews, the more representative it is of the opinions issued about the product.

Given an adjacency matrix W' which represents a graph of reviews that are con-

nected due to a minimum similarity threshold, the centrality scores are iteratively calculated using PageRank. The centrality indices obtained represents a ranking of vertices' importance, used to indicate the ranking of most relevant reviews.

2.3 Experiment Design

In this section, the experimental setting used to evaluate MRR is detailed. Also, the dataset used is described, as well as the methods employed as baseline for comparison, the helpfulness definition adopted as Gold-standard, and the metric applied for relevance ranking evaluation. Finally, some details on other experiments performed to assess MRR are presented.

2.3.1 Datasets

The dataset comprises reviews of electronic products and books extracted from the Amazon website (MCAULEY; PANDEY; LESKOVEC, 2015). The sampling was obtained using the same methodology proposed by Chua et al. (2016), which used only products with more than 30 reviews, and reviews longer than three sentences and with more than five users' helpfulness votes. The resulting datasets contain 19,756 reviews of electronics products and 24,234 reviews of books. For each review, the following features were used: a) review rating; b) review text; and c) the number of positive and negative votes. Table 5.1 describes the profiling of the datasets.

Table 2.1: Profiling of the Amazon dataset.

	Electronics	**Books**
Votes	48.20 ($\pm$ 302.84)	29.71 ($\pm$ 73.58)
Positive	40.12 ($\pm$ 291.99)	20.60 ($\pm$ 64.18)
Negative	8.08 ($\pm$ 22.27)	9.11 ($\pm$ 21.44)
Rating	3.73 ($\pm$ 1.50)	3.41 ($\pm$ 1.54)
Words	350.32 ($\pm$ 402.02)	287.44 ($\pm$ 273.75)
Products	383	461
Total	19,756	24,234

2.3.2 Baseline

As baseline the two state-of-the-art unsupervised ranking methods described in Section 5.6 (TSUR; RAPPOPORT, 2009; WU; XU; LI, 2011) were adopted. Additionally, to measure the gap between MRR and a supervised one, the results were compared with two regression Support Vector Machine (SVM) regression models with 10-fold cross validation. The SVM models were trained using two distinct set of features: a) textual features represented by TF-IDF and the star score; and b) the same features used by Wu et al. (2011), who also made this comparison, namely review length, counts of POS-category (i.e. adjective, noun, verb, and adverb), number of sentences and uni-gram counts.

2.3.3 Gold Standard and Evaluation Metric

Previous work on automatic review helpfulness assessment (KIM et al., 2006; MUDAMBI; SCHUFF, 2010) model the relevance as the proportion of users who have given a positive vote about the total of given votes (positive and negative votes). Thus, the evaluating of how MRR mimics the human model of relevance is measuring in terms of helpfulness. The Gold Standard is given by Equation 2.5.

$$h(r \in R) = \frac{vote_{+}(r)}{vote_{+}(r) + vote_{-}(r)} \tag{2.5}$$

where r is a review belonging to a set of reviews R, $votes_{+}(r)$ is the number of people who find r useful and $votes_{-}(r)$ is the number of users that do not consider it useful.

2.3.4 Normalized Discounted Cumulative Gain

Since each product that composes the gold standard has a distinct helpfulness score, Normalized Discounted Cumulative Gain (NDCG) (JÄRVELIN; KEKÄLÄINEN, 2002) measure were employed as evaluation metric of MRR's ranking task. NDCG is a metric used in information retrieval, which measures the performance of a recommendation system based on the graded relevance of the recommended entities. NDCG emphasizes the head of the list, where NDCG@k represents by the optimal ranking for only the top-k items. It varies from 0.0 to 1.0, with 1.0 representing the ideal ranking of the entities. The MRR was assessed using the top review (NDCG@1) and the top-5 reviews

(NDCG@5).

2.3.5 MRR Parameterization

The proposed algorithm relies on two parameters (α, β) that change its behavior and the quality of results. To evaluate the sensibility of MRR to these parameters, an optimization method to find the best values based on the minimization of Mean Squared Error were used. The results obtained from the optimization method were compared with the results obtained from the manually defined values for α and β.

The method used finds the best α and β, was Newton-Conjugate-Gradient, which is numerical optimization technique that minimizes functions of multiples variables. It was employed the entire set of reviews described in Section 4.1, with the intent of finding a single pair of parameters that maximize the ranking quality (measured using NDCG@5) for both electronics and books reviews. The best parameters found were $\alpha = 0.867168$ and $\beta = 0.85533$.

2.3.6 Graph-Specific Threshold Assessment

To assess the contribution of the Graph-Specific Threshold proposed to MRR, the NDCG results obtained using different similarity thresholds manually set were compared with the ones produced using the Graph-Specific threshold. These were calculated according to the optional α and β parameters, as defined in the previous subsection.

2.3.7 Run-time Evaluation

Since the MRR algorithm was designed to deal with a large number of reviews, a run-time assessment was also performed. The analysis was limited to the electronic products dataset. For each product in this dataset, the time to produce the relevance ranking according to MRR and the unsupervised baselines was measured. The assessment was performed using i7 1.8 GHz Intel machine with 4Gb of RAM.

2.4 Results and Discussions

This section discusses the evaluation of MRR with regard to the adopted baselines in terms of helpfulness ranking assessment using reviews of two different domains. It also discusses the results of the other assessments performed, namely parameters' sensitivity, the contribution of the graph-specific threshold, and run-time performance.

2.4.1 Relevance Ranking Assessment

Tables 2.2 and 2.3 display the mean NDCG (it has an NDCG value for each product in the review set) for the electronics and books datasets, where the baselines are referred to as REVRANK (TSUR; RAPPOPORT, 2009), PR_HS_LEN (WU; XU; LI, 2011), and the supervised SVM regression models as SVM_TFIDF (trained using TF-IDF vectors extracted from the reviews) and SVM_WU (using the same features used by Wu et al. (2011)).

Table 2.2: Mean Performance on Book Reviews

	NDCG@1	NDCG@5
SVM_WU	0.80770	0.91817
SVM_TFIDF	**0.85539**	**0.93119**
REVRANK	0.66052	0.68172
PR_HS_LEN	0.72689	0.77131
MRR	0.79877	0.81876

Table 2.3: Mean Performance on Electronic Reviews

	NDCG@1	NDCG@5
SVM_WU	0.76416	0.91535
SVM_TFIDF	0.88986	**0.94621**
REVRANK	0.67903	0.72133
PR_HS_LEN	0.87434	0.87184
MRR	**0.89403**	0.89246

The complete distributions are depicted in the boxplots of figures 2.2 and 2.3.

The mean NDCG results in tables 2.2 and 2.3 show that MRR outperformed all unsupervised baselines. For the book dataset, the differences range from 4.7 to 7.2 percentage points (pp) when compared to the runner-up method, namely PR_HS_LEN. For

Figure 2.2: Distribution of results obtained in MRR and the baseline on books reviews.

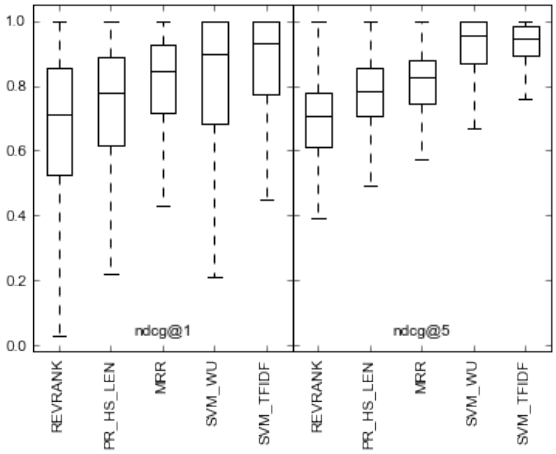
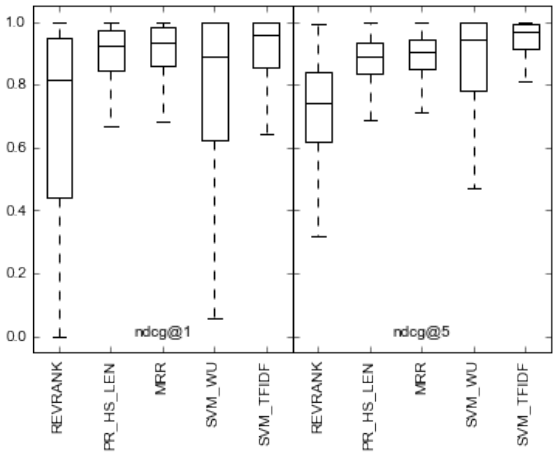


Figure 2.3: Distribution of results obtained from MRR and the baseline on electronics reviews.



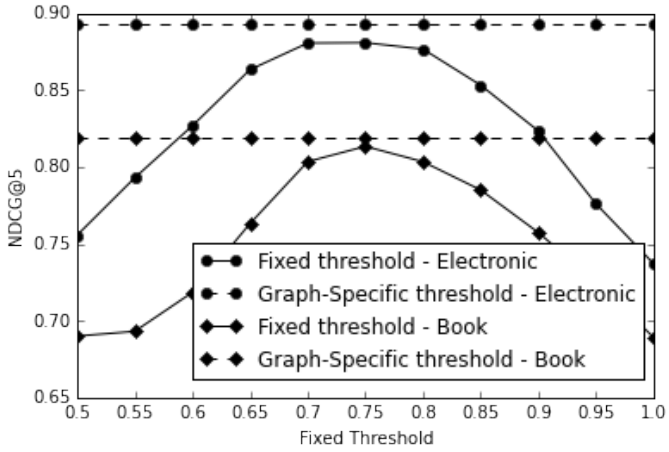
the electronic dataset, the difference is approximately 2 pp in all cases. The boxplots show that MRR results are not only better in average, but also in terms of lower and upper quartiles, minimum and maximal values. The Wilcoxon statistical test (WILCOXON; KATTI; WILCOX, 1970) with a significance level of 0.05 verified that MRR results are statistically superior, except in a single case, namely NDCG@1 in the electronic dataset.

With regard to the supervised models, MRR yields comparable results to both SVM_TFIDF and SVM_WU at NDCG@1, which is an excellent result for an unsupervised model. In the books dataset, the difference in favor of SVM_TFIDF and SVM_WU is 5.6 pp and 0.9 pp, respectively. In the electronics dataset, MRR outperforms both models (0.42 pp and 13 pp, respectively). The Wilcoxon statistical test, with significance level 0.05, verified that MRR NDCG@1 results on electronics dataset at NDCG@1 are statistically superior, and comparable in the other cases. However, using NDGC@5, both SVM models outperforms MRR, and the difference is statistically significant.

2.4.2 Graph-Specific Threshold Assessment

Figure 2.4 displays the mean NDCG@5 for both electronics and books datasets, using a fixed threshold (FT) and a graph-specific threshold (GST). It is possible to see that in both datasets, MRR performance is always better using a Graph-Specific threshold. Thus, in addition to eliminating the burden of experimenting with different thresholds, this approach yields the best results. Considering the manually set thresholds FT, the best NDCG result was found for similarity threshold 0.75 for both books and electronic datasets. Nevertheless, these results are, respectively, 6.1 pp and 1.1 pp inferior to the ones obtained using graph-specific thresholds.

Figure 2.4: Graph-Specific Threshold versus different values for Fixed Thresholds.



2.4.3 Parameter Sensitivity Assessment

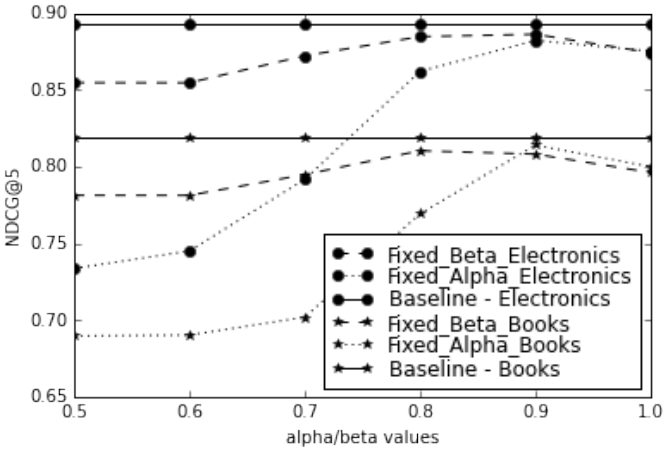
Given the optimal parameters ($\alpha = 0.867168$ and $\beta = 0.85533$), the impact of a manual parametrization on the quality of MRR's ranking using different values of $alpha$ and $beta$ on both electronic and books reviews were measured, as follow:

- **Fixed_Beta_Electronics:** β=0.85533 and $\alpha \in [0.5, 1]$ on electronics reviews;
- **Fixed_Alfa_Electronics:** α=0.867168 and $\beta \in [0.5, 1]$ on electronics reviews;
- **Fixed_Beta_Books:** β=0.85533 and $\alpha \in [0.5, 1]$ on books reviews;
- **Fixed_Alfa_Books:** α=0.867168 and $\beta \in [0.5, 1]$ on books reviews.

Results are depicted in Figure 2.5. Compared to the mean NDCG@5 scores using the optimal parameters, the experiment shows that while α in all settings had a low influence (0.7%-4%) on the results, β produced the highest performance variation (1%-17%

on electronics reviews and 0.5%-15% on books reviews). However, the results showed that high values for *beta* tend to produce a good mean NDCG score. Nevertheless when $0.8 \leq \beta \leq 0.9$, the MRR algorithm produces stable results, varying only 3% for electronics and 6% for books compared which is not a significant difference between the optimal parameters.

Figure 2.5: Influence of MRR's parameters on NDCG results



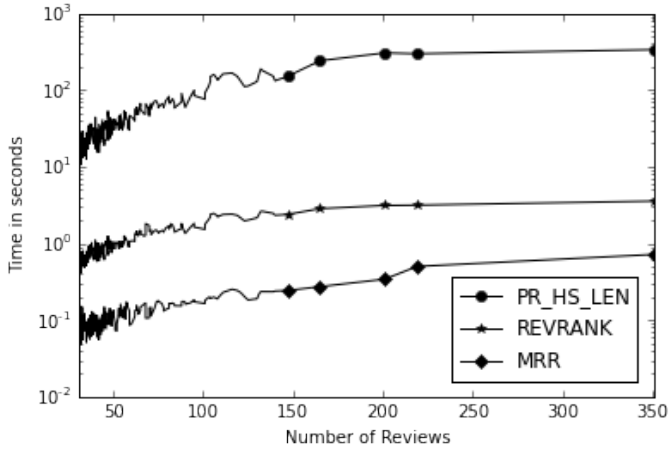
2.4.4 Run-time Assessment

Figure 2.6 shows the time required for producing a ranking for each one of the 383 products in the electronic dataset using MRR, PR_HS_LEN and REVRANK (log scale). The number of reviews per product ranges from 30 up to 350. Although all approaches present a similar growth rate, MRR presents a significantly lower running time, thus indicating that it is more suitable than the baselines to process a larger number of reviews.

2.5 Related Work

Several studies have addressed the task of predicting the relevance of reviews. The use of sentiment analysis as the basis for predicting review relevance using supervised learning has been addressed by works such as (MUKHERJEE; LIU, 2012; TANG; QIN; LIU, 2015), but the relationship between sentiment and relevance has not been proved consistent (CHUA; BANERJEE, 2016). Using regression techniques, Chua et al. in (CHUA; BANERJEE, 2016) explored other features in addition to sentiment to predict

Figure 2.6: Run-time comparison between MRR, REVRANK and PR_HS_LEN for electronic products reviews.



relevance, such as text properties (e.g. number of words), product type (search and experience) and self-claimed expertise, concluding that the best predictors were the number of words and product key aspects. Another study (WOLOSZYN; SANTOS; WIVES, 2016) explored the readability aspects of helpful reviews, concluding there is a high correlation between several intelligibility indicators and usefulness.

Relevance is often measured in terms of helpfulness, interpreted as the perceived value of a given entry to support purchase decisions (MUDAMBI; SCHUFF, 2010). Using votes given by users to a question such as “Was this review helpful to you?", many studies try to predict review’s relevance (XIONG; LITMAN, 2011; ZENG; WU, 2013; YANG et al., 2015; KIM et al., 2006), always using supervised learning strategies. Regression algorithms consistently improved the prediction of helpfulness in the context of cross-language reviews (WAN, 2013). However, approaches based on users’ votes are affected by the “cold start problem" (LAM et al., 2008).

A common drawback of supervised learning approaches is that the quality of results is heavily influenced by the availability of a large, domain-dependent annotated corpus to train the model. Unsupervised learning techniques are attractive because they do not imply the cost of corpus annotation.

RevRank (TSUR; RAPPOPORT, 2009) is a fully unsupervised approach to rank reviews using the word frequency distribution of a given review dataset. First, it creates a core virtual review using the optimal 200 most frequent words. Then it ranks the reviews based on the distance from this virtual review. The ranking results are inferior compared to centrality-based techniques.

Wu et al. proposed a method in (WU; XU; LI, 2011) to rank relevant reviews

which combine the centrality scores assigned to individual sentences and the overall reviews' length to produce an overall centrality score for reviews. It employs PageRank twice: a) to compute the centrality scores of sentences based on similarity of specific POS-tagged features (i.e. nouns, verbs, and adjectives), and b) to compute the centrality of reviews leveraging the scores of individual sentences and review length. Although it achieves results that are only slightly inferior compared to supervised learning methods, its run-time performance is affected by the design choices: a) the review similarity function that requires previous computation of sentence similarity using centrality; b) double use of PageRank; and c) extraction of specific POS-tagged features for similarity computation, which increases processing time and its error prone due to issues in user-generated content.

In short, MRR combines the core virtual review concept (TSUR; RAPPOPORT, 2009) with review relevance by centrality (WU; XU; LI, 2011), powered by a graph-specific similarity threshold.

2.6 Chapter summary

The MRR is a novel unsupervised algorithm that identifies relevant reviews based on the concept of node centrality. The intuition behind MRR is that central reviews highlight aspects of a product that many other reviews frequently mention, with similar opinions, as expressed in terms of ratings.

The proposed algorithm requires no prior domain knowledge and no training data, which avoids costly and error-prone manual training annotations. Compared to related work, MRR: a) outperforms baseline unsupervised techniques (WU; XU; LI, 2011; TSUR; RAPPOPORT, 2009) imitating the user vote model, and has a comparable performance with regard to a supervised regression model (in a specific setting); b) presents better run-time performance due to a computationally inexpensive reviews similarity function; c) is adaptable to the characteristics of the reviews dataset by setting a specific similarity threshold to each product's sets of reviews.

In the experiments, MRR showed to be suitable for products such as books, on which opinions can be highly open-ended, and electronics, which have a relatively small number of well-defined features (TSUR; RAPPOPORT, 2009). In addition, the graph-specific threshold achieves the best results adapting itself to the characteristics of the reviews set, and eliminates the burden of experimenting with different thresholds. The

assessment of the sensitivity for the α and β parameters showed that the latter has the stronger influence. Nevertheless, there is not a significant difference between the optimal parameters, specially the ones set in the range of [0.8-0.9].

In terms of run-time cost, MRR is computational inexpensive when compared to other graph-centrality methods that are based on sentence similarity, since it is based on TF-IDF and stars features of reviews to compute the review centrality in a graph. Such a feature allows MRR to process a large number of reviews in a shorter time lapse than the baselines.

2.7 Final Remarks

This Chapter presented the work carried out to rank reviews by their relevance (RQ1). Furthermore, in our experiments (better discussed in Chapter 8), we observed that the Most Helpful Review, usually, do not cover most of the user interests. Nonetheless, I believe that combining MRR with a biased coverage of the user's interests (RQ2) can generate useful summaries to users. In this sense, next Chapter presents a research addressed to biased automatic text summarization system.

3 BIASED SUMMARIZATION

Automatic Text Summarization are systems build to extract the most important passages from a text, in other hand, a biased summary can covers a specific set of subjects. This Chapter presents BEATnIk, an algorithm to generate biased summaries, that cover different set of subjects which is not necessarily the most important. In this sense, BEATnIk is an step towards answering the Research Question 2 - How to create a textual summary which covers the desirable information for a specific user.

Similarly to the previous chapter, it also use reviews to validate the experiments, however for a different purpose: extracts from user's review educational aspects of a movies. It is important to state that BEATnIk was developed to created not only to generate summaries which covers educational aspects of movies, but also it is capable to generate reviews which covers the information need by a specific user. The central content of this chapter was published at Brazilian Symposium on Computers in Education[1], and have received a mention of honor.

3.1 Introduction

The use of extracurricular learning material is a common practice inside a classroom. Teachers have been increasingly using movies, software and other kinds of learning objects that can support the teaching of the class subject, and some examples of such practices can be found in (GIRAFFA; MULLER; MORAES, 2015; OLIVEIRA; RODRIGUES; QUEIROGA, 2016; CASTRO; WERNECK; GOUVEA, 2016). The use of movies is one of the simplest ways to support teaching because it is easily available and is a time-controlled experience inside the classroom. In this sense, Websites such as TeachWithMovies[2], arise as a valuable support to the creation of lesson plans. In this website, a set of movies is described by teachers to be used as learning objects inside a classroom. Each movie description contains at least the movie's benefits and possible problems, a helpful background, a discussion; besides, with some descriptions, there are also questions to be used in class. The preparation of such type of material is a time-consuming

[1]Complete Reference: Woloszyn, V., Machado, G. M., de Oliveira, J. P. M., Wives, L., Saggion, H. (2017, October). BEATnIk: an algorithm to Automatic generation of educational description of movies. In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 28, No. 1, p. 1377).

[2]http://www.teachwithmovies.org/

activity, and an educational summary can help in the elaboration of a longer movie-based lesson plan.

Several works address the challenge of extracting specific aspects from users' reviews to compose a summary about a movie or a product. Most of those works rely on supervised algorithms such as classification and regression (XIONG; LITMAN, 2011; ZENG; WU, 2013; YANG et al., 2015; WOLOSZYN et al., 2017a). However, the quality of results produced by supervised algorithms is dependent on the existence of a large, domain-dependent training dataset. In this sense, semi-supervised and unsupervised methods are an attractive alternative to avoid the labor-intense and error-prone task of manual annotation of training datasets.

Considering such context this work describes BEATnIk (Biased Educational Automatic Text Summarization), which is an unsupervised algorithm to generate biased summaries that cover educational aspects of movies from users' reviews. BEATnIk can help teachers in providing educational descriptions for movies. So, the work's main contributions are: a) the description of a tool to assist professors in the creation of lesson plans from the movies' reviews; and b) an unsupervised algorithm which outperforms the baseline, imitating the human educational description of the movie. BEATnIk can also be employed in other domains, it would only require small modifications to be able to generate, for instance, a biased summary that covers the personal user's aspect of interest about products on Online Collaborative Review Websites. It is important to highlight that BEATnIk is open source and it is available on the Internet [3].

The rest of this Chapter is organized as follows. Section 5.6 discusses the related works. Section 3.2 present the datasets employed on this work. Section 3.3 present details of BEATnIk algorithm. Section 5.4 describes the design of our experiments, and Section 3.5 discusses the achieved results. Section 5.7 summarizes our conclusions and presents future research directions.

3.2 Datasets Employed

As the goal of our approach was to build a biased summarizer for educational purposes, this work employed two datasets to perform the experiments. The first served as a word thesaurus to implement the educational bias, and it was collected from an educational website TeachWithMovies (TWM) where a set of movies are described by teachers

[3] http://xx.yy.zz

with the goal to use them as learning objects inside a classroom. The second dataset is Amazon Movie Reviews (AMR) (MCAULEY; LESKOVEC, 2013) which provides user comments about a large set of movies. Since only the movies that appeared in both datasets could be used, a filter was applied, which ended up with 256 movies to perform our evaluation. Next Section describes with more details each dataset.

3.2.1 Teaching with Movies

The TeachWithMovies dataset was collected through a crawler developed by us. Different teachers described the movies on the website, but each movie has only one description, this was a challenge while collecting the data because the information was not standardized or had associated metadata.

However, is important to noticed that some movies presented common information: i) movie description; ii) rationale for using the movie; iii) movie benefits for teaching a subject; iv) movie problems and warnings for young watchers; and v) objectives of using this movie in class. The developed crawler extracted such information, and the movie description was used since it contains the greatest amount of educational aspects. In the end, 408 unique movies and video clips were extracted, but after matching with the Amazon dataset, only 256 movies were used.

3.2.2 Amazon Movie Reviews

The Amazon Movie Reviews was collected with a timespan of more than ten years and consists of proximately 8 millions of reviews that include product and user information, ratings, and a plain text review. In Table 3.1 is shown some statistics about the data.

Table 3.1: Amazon Movie Reviews Statistics

Dataset Statistics	
Number of reviews	7,911,684
Number of users	889,176
Expert users (with >50 reviews)	16,341
Number of movies	253,059
Mean number of words per review	101
Timespan	Aug 1997 - Oct 2012

3.3 BEATnIk Algorithm

In BEATnIk, a complete graph is constructed for each movie. In this graph, each sentence extracted from the Amazon's dataset becomes a node, and each edge's weight is defined by a similarity measure applied between sentences. An adapted cosine equation assesses the similarity. The algorithm then employs PageRank (PAGE et al., 1999) to compute the centrality of each node. The intuition behind this approach is that central sentences highlight aspects frequently mentioned in a text. Also, BEATnIk takes into account keywords extracted from the lesson plans of TWM (used as a bias) to compute the importance of each sentence. The final educational summary is based on the centrality score of the sentences weighted by the presence of educational keywords.

Let S be a set of all sentences extracted from the R user's reviews about a single movie, BEATnIk builds a graph representation $G = (V, E)$, where $V = S$ and E is a set of edges that connect pairs $\langle u, v \rangle \in V$. The score of each node (that represent a sentence) is given by the harmonic mean between its centrality score on the graph given by PageRank, and the sum of the frequencies of its education keywords (stated in equation 3.2). The pseudo-code of BEATnIk is displayed in Algorithm 4, where G is represented as the adjacency matrix W.

Algorithm 2 - BEATnIk Algorithm (S,B): O

- Input: a set of sentences extracted from the Amazon's reviews R, and a corpora B used as bias and

- Output: a extractive biased summary O based on reviews R.

1: **for** each $u, v \in S$ **do**
2: $W[u,v] \leftarrow$ *idf-modified-cosine*(u,v)
3: **end for**
4: **for** each $u, v \in S$ **do**
5: **if** $W[u,v] \geq \beta$ **then**
6: $W'[u,v] \leftarrow 1$
7: **else**
8: $W'[u,v] \leftarrow 0$
9: **end if**
10: **end for**
11: $P \leftarrow PageRank(W')$
12: **for** each $u \in S$ **do**
13: $K \leftarrow$ *sim-keyword*(u, B)
14: $O[u] \leftarrow \frac{\|S\| P_u K}{P_u + K}$
15: **end for**
16: Return O

The main steps of the BEATnIk algorithm are: (a) it builds a similarity graph (W) between pairs of reviews of the same product (lines: 1-3); (b) the graph is pruned (W') by removing all edges that do not meet a minimum similarity threshold, given by the parameter β[4] (lines 4-10); (c) using PageRank, the centrality scores of each node is calculated (line 11); (d) using the educational corpora, each sentence is scored according the presence of educational keywords (line 13); (e) The final importance score of each node is given by the harmonic mean between its centrality score on the graph, and the sum of its education keywords frequencies (line 14).

To get the similarity between two nodes, it uses a metric that is an adapted cosine

[4]The best parameter obtained in our experiments is $\beta = 0.1$

difference of the two corresponding sentence vectors (ERKAN; RADEV, 2004):

$$\text{idf-modified-cosine}(x,y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}} \tag{3.1}$$

where $\text{tf}_{w,s}$ is the number of occurrences of the word w in the sentence s. It uses the approach described in (MIHALCEA; TARAU, 2004) to extract the keywords from the educational corpora. The similarity between the sentences and the keywords extracted from the TWM lesson plans are given by the following equation:

$$\text{sim-keyword}(x,B) = \sum_{w \in x} \text{tf}_{w \in keywords(B)} \tag{3.2}$$

The comparison of our approach to TextRank (MIHALCEA; TARAU, 2004), which is also a Graph-based Automatic Text Summarization, revealed that BEATnIk generates summaries closer to the educational description of the movies in TWM (details are presented in the next section).

3.4 Experiment Design

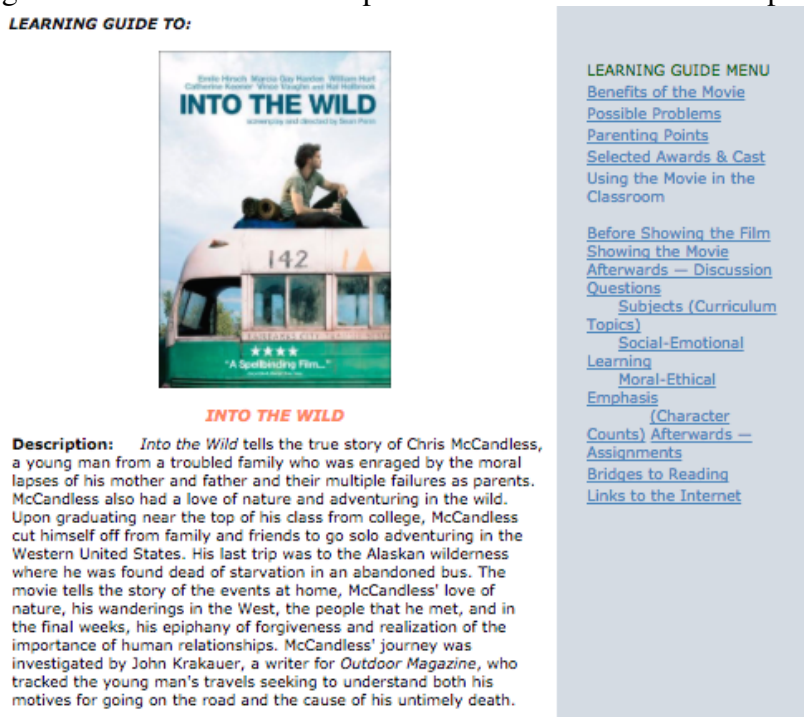
This section presents the experimental setting used to evaluate BEATnIk. It describes the method employed as the baseline for comparison, the educational plans adopted as Gold-standard and the metric applied for evaluation, as well as details of the experiment, performed to assess BEATnIk.

3.4.1 The baseline

The results obtained from our proposed approach are compared with Textrank (MIHALCEA; TARAU, 2004) algorithm. Textrank was chosen because it is also a graph-based ranking algorithm and has been widely employed in Natural Language tools (ŘEHŮŘEK; SOJKA, 2010).

Textrank essentially decides the importance of a sentence based on the idea of "voting" or "recommending". Considering that in this approach each edge represents a vote, the higher the number of votes that are cast for a node, the higher the importance of the node (or sentence) in the graph. The most important sentences compose the final

Figure 3.1: A summarized snapshot of “Into the Wild” lesson plan



LEARNING GUIDE TO:

INTO THE WILD

INTO THE WILD

Description: *Into the Wild* tells the true story of Chris McCandless, a young man from a troubled family who was enraged by the moral lapses of his mother and father and their multiple failures as parents. McCandless also had a love of nature and adventuring in the wild. Upon graduating near the top of his class from college, McCandless cut himself off from family and friends to go solo adventuring in the Western United States. His last trip was to the Alaskan wilderness where he was found dead of starvation in an abandoned bus. The movie tells the story of the events at home, McCandless' love of nature, his wanderings in the West, the people that he met, and in the final weeks, his epiphany of forgiveness and realization of the importance of human relationships. McCandless' journey was investigated by John Krakauer, a writer for *Outdoor Magazine*, who tracked the young man's travels seeking to understand both his motives for going on the road and the cause of his untimely death.

LEARNING GUIDE MENU
Benefits of the Movie
Possible Problems
Parenting Points
Selected Awards & Cast
Using the Movie in the Classroom

Before Showing the Film
Showing the Movie
Afterwards — Discussion Questions
Subjects (Curriculum Topics)
Social-Emotional Learning
Moral-Ethical Emphasis
(Character Counts)
Afterwards — Assignments
Bridges to Reading
Links to the Internet

summary.

3.4.2 Gold-Standard

The lesson plans found on the TWM website were used as a gold-standard to assess BEATnIk summaries. An English-speaking teacher describes each lesson plan and takes into consideration the educational aspects of the movie.

The lessons are categorized by movie genre, learning discipline, recommended age (from 3 years-old to college level), and alphabetical order. Inside the lesson plans, there is also some learning goals regarding the movie, such as the learning subject, the social-emotional learning, and the ethical emphasis.

Taking, for instance, the summary of “Into the Wild” lesson plan presented in Figure 3.1, where a teacher highlighted the importance of human relationships. At the top right, it is found the structure of the whole lesson available online [5]. In the remaining of the lesson, the teacher still presents some benefits of the movie, such as risky behavior can have fatal consequences and relationships with people are an essential part of life.

TWM provided a well-described educational dataset, and despite the lack of standardization of lessons plans, this work used it successfully as a gold-standard to perform

[5] http://www.teachwithmovies.org/guides/into-the-wild.html

our experiments.

3.4.3 Evaluation Metric

The evaluation was performed by applying ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (LIN, 2004), which is a metric inspired on the Bilingual Evaluation Understudy (BLEU) (SAGGION; POIBEAU, 2013).

Specifically, ROUGE-N were used in the evaluation, which makes a comparison of n-grams between the summary to be evaluated and the “gold-standard" (in our case, BEATnIk summaries and TWM lesson plans, respectively). Only the first 100 words of the summaries of BEATnIk’s and the baseline’s summary were considered, since it corresponds to the median size of the gold-standard. ROUGE was chosen because it is one of most used measures in the fields of Machine Translation and Automatic Text Summarization (POIBEAU et al., 2012).

3.4.4 BEATnIk’s bias

The set of lesson plans extracted from TMW was used as an educational bias for BEATnIk algorithm. When generating a biased summary for a specific movie, BEATnIk does not take in consideration such movie lesson plan. Instead, it builds a graph using all other movies information, excepting the movie to be summarized. This strategy avoids any positive influence on the performance of the predictive model.

The retrieved corpus was composed of 991 sentences and 2,811 unique tokens. Table 3.2 describes the first 20 keywords extracted from TWM corpus.

3.5 Results

This section presents the BEATnIk’s evaluation regarding the adopted baselines concerning precision, recall, and f-Score obtained by using ROUGE-N.

The gold-standard utilized in the experiments, as already stated in Section 5.4, is the educational description extracted from the TWM website. Table 3.3 shows the mean Precision, Recall, and F-Score, considering both BeatnIk and Textrank (the gold-standard used as the baseline).

Table 3.2: Keywords extracted from the lesson plans in TWM

Keywords	Frequency	Keywords	Frequency
film	0.01390	class	0.00354
movi	0.01062	famili	0.00345
children	0.00475	bulli	0.00345
benefit	0.00457	parent	0.00336
father	0.00440	boy	0.00319
use	0.00414	help	0.00311
stori	0.00406	point	0.00311
discuss	0.00388	live	0.00285
question	0.00362	life	0.00276
child	0.00362	time	0.00276

The results presented in Table 3.3 show that BEATnIk outperformed the baseline in all measurements carried out. Regarding Precision, the differences range from 4.9 to 11.9 percentage points (pp) on all ROUGE-N analyzed, where N is the size of the n-gram used by ROUGE. The Wilcoxon statistical test, with a significance level of 0.05, verifies that BEATnIk is statistically superior when compared to the baseline. Regarding recall, the differences are also in favor of BEATnIk, ranging from 4.7 to 11.5 pp when compared to the baseline.

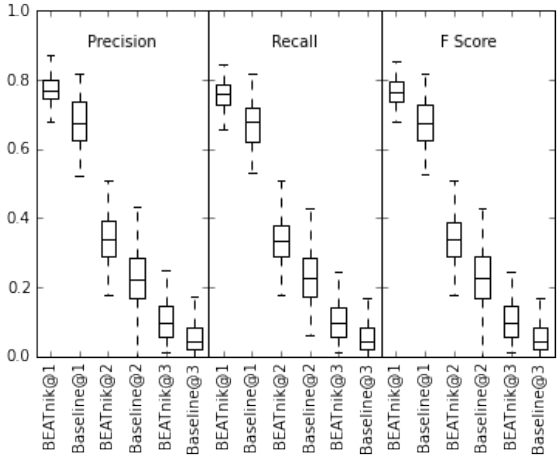
Table 3.3: Mean of ROUGE results achieved by BEATnIk and the Baseline

ROUGE-n	*Baseline*	*BEATnIk*	p-values
Precision-1	0.65615	**0.77028**	< 0.05
Recall-1	0.65003	**0.75611**	< 0.05
F_score-1	0.65283	**0.76296**	< 0.05
Precision-2	0.22394	**0.34350**	< 0.05
Recall-2	0.22192	**0.33744**	< 0.05
F_score-2	0.22284	**0.34037**	< 0.05
Precision-3	0.06313	**0.11268**	< 0.05
Recall-3	0.06387	**0.11102**	< 0.05
F_score-3	0.06347	**0.11182**	< 0.05

Regarding the distribution of Rouge's results, the boxplot showed in Fig 3.2 indicates that BEATnIk results are not only better in mean, but also concerning lower and upper quartiles, minimum and maximal values.

To illustrate the differences between the BEATnIk and a generic text summarizer

Figure 3.2: Distribution of Rouge results.



on the task of extracting the educational aspects from the movie's reviews, consider the snippet of summaries about the movie 'Conrack' at table 3.4. In this example, while BEATnIk highlights the educational aspects such as *method lesson, teaching, and children*, the generic text summarizer used as baseline highlights the aspects frequent mentioned in the reviews, such as related to the *screenplay* and the *director*.

Table 3.4: Snippets of the summaries generated by BEATnIk and the Baseline about movie 'Conrack'

BEATnIk	**Baseline**
As well as being a method lesson in teaching, it is also a good personal film, and even if you don't warm to Jon Voight's character immediately, you will love the little children. [...]	*The director achieved a glimmering one in this hidden gem adapted from author Pat Conroy's novel The Water Is Wide.* [...]

3.6 Related work

Automatic Text Summarization (ATS) techniques have been successfully employed on user-content to highlight the most relevant information among the documents (ERKAN; RADEV, 2004; RADEV et al., 2004; GANESAN; ZHAI; HAN, 2010; SAGGION; POIBEAU, 2013). Regarding the techniques employed, some works have explored unsupervised

methods based on graph centrality. In RevRank (TSUR; RAPPOPORT, 2009) it is presented the concept of Virtual Core Review, which is a graph composed by dominant terms of the documents, where the relevance of each document is given according to their distance from the “Virtual Core" document. Another example is presented in (WU; XU; LI, 2011) that combines the centrality scores of each sentence with the documents' length to produce an overall centrality score for each review. However, this method does not scale well due to the chosen centrality granularity, which implies the dual use of PageRank, and requires pre-processing to identify specific textual features (e.g. nouns, adjectives).

There are also studies using supervised learning strategies to predict the text relevance (XIONG; LITMAN, 2011; ZENG; WU, 2013; YANG et al., 2015; WOLOSZYN et al., 2017a). Additionally, the use of regression algorithms consistently improves the prediction of helpfulness (WAN, 2013). However, a common drawback of supervised learning approaches is that the quality of results is heavily influenced by the availability of a large, domain-dependent annotated corpus to train the model. Unsupervised learning techniques are attractive because they do not imply the cost of corpus annotation either training. Therefore, it is described in this work an unsupervised biased algorithm to extract educational aspects from movies' reviews with the goal to assist teachers on the task of creating movie-based lesson plans.

3.7 Chapter summary

In this chapter BEATnIk, an algorithm to generates biased summaries based on the concept of node centrality, was presented. The intuition behind BEATnIk is that central sentences containing “educational keywords" that are extracted from users' reviews about movies; are closer to a human-made educational description of the movie, than a general summary. This work proved this assumption and showed that BEATnIk achieved statistically superior results than Textrank.

The main contributions are the design and presentation of BEATnIk as a tool to assist professors in the creation of lesson plans based on movies' reviews, and a methodology designed to assess BEATnIk, which outperformed the baseline, imitating the human educational description of the movies.

3.8 Final Remarks

It is important to state that it was found out a considerable number sentences with repeated information (e.g., similar sentences that express the same idea) and helpful sentences with low centrality indexes that lead us to consider the investigation of other techniques to select the most relevant sentences to compose a review. Next Chapter will present an approach to spot fake content.

4 RANKING DOCUMENTS BY TRUST

This chapter presents a semi-supervised approach to spot fake content, namely DistrustRank. The detection of fake content described in this work plays an important role for both Research Question 1 and 2, since identifying trustful information is essential to ensure that only reliable information is included in the summary for the user.

For validating our proposed algorithm, we employed DistrustRank for detecting fake news. Although News and Reviews have different characteristics, the model here proposed takes into consideration only textual similarities, which can be applied in both domains. The work described here was published at 10th ACM Conference on Web Science, Amsterdam, 27-30 May 2018[1].

4.1 Introduction

Many people have access to the News through different online information sources, ranging from search engines, digital forms of mainstream News channels to social network platforms. Compared with traditional media, information on the Web can be published quickly, but with few guarantees on the trustworthiness and quality. This issue can be found in different domains, such as fake reviews on collaborative review Websites, manipulative statements about companies, celebrities, and politicians, among others (GUPTA et al., 2013; LI et al., 2012).

The task of assessing the believability of a claim is a thorny issue. Kumar's work (KUMAR; WEST; LESKOVEC, 2016) reports that even humans sometimes are not able to distinguish hoax from authentic ones and that quite a few people could not differentiate satirical articles from the true News (e.g., http://www.nypost.com/2018/02/01/mom-teams-up-with-daughter-to-fight-girl-on-school-bus/). With the increasing number of hoaxes and rumors, fact-checking Websites like *http://snopes.com/*, *http://politifact.com/*, and *http://fullfact.org/*, have become popular. These Websites compile articles written by experts who manually investigate controversial claims to determine their veracity, providing shreds of evidence to for the verdict (e.g., true or false).

Many works have addressed the problem of false claims detection. Most of them rely on supervised algorithms such as classification and regression models (DORI-HACOHEN;

[1] Woloszyn, V., & Nejdl, W. DistrustRank: Spotting False News Domains. In: Proceedings... ACM Conference on Web Science, 10th. 2018. pp. 221-228.

ALLAN, 2013; RAJKUMAR et al., 2014; KUMAR; WEST; LESKOVEC, 2016; POPAT et al., 2016; SHARIFF; ZHANG; SANDERSON, 2017; STANOVSKY et al., 2017; HORNE; ADALI, 2017) which rely on annotated data sets for the training step. Thus, as mentioned before, this thesis aims to leverage unsupervised methods to avoid the labor-intense and error-prone task of manual annotation of training data sets.

In this Chapter DistrustRank is presented; it is a semi-supervised algorithm that identifies unreliable News Websites based only on the headline extracted from the News article's link. This is proposed because in News Websites articles are generally shared using a long link that contains the news headline and acts as a good summary of the article content. This choice is motivated by performance issues, since for a fast and scalable method the extraction of features for comparison cannot be time-consuming. Additionally, using only links instead of entire News article content is a good strategy to help the integration of DistrustRank with search engines since it does not need additional features. The use of links as the main feature is also a common strategy in other areas, such as Query Re-Ranking (BAYKAN; HENZINGER; WEBER, 2013; SOUZA et al., 2015).

DistrustRank constructs a weighted graph where nodes represent Websites, connected by edges based on a minimum similarity between a pair of Websites, and then compute the centrality using a biased PageRank, where a bias is applied to the selected set of seeds. In addition, DistrustRank takes into account fake Websites similarities, as a minimum similarity threshold is dynamically defined based on the characteristics of the set of false Websites. The resulting graph is composed of several components, where each component represents Websites with similar characteristics. Next, a search that begins at some particular node v will find the entire connected component containing v. Finally, the centrality index of the neighbors of v is used to compose the final distrust rank.

The output of the method presented in this Chapter is a trust (or distrust) rank that can be used in two ways:

1. as a counter-bias to be applied when News about a specific subject is ranked, in order to discount possible boosts achieved by false Websites;
2. to assist people to identify sources that are likely to be fake (or reputable), suggesting which Websites should be examined more closely or to be avoided.

The experiments on Websites indexed by Internet Archive[2] reveal that DistrustRank outperforms the chosen supervised baseline (Support Vector Machine) in terms of imitat-

[2]http://web.archive.org/

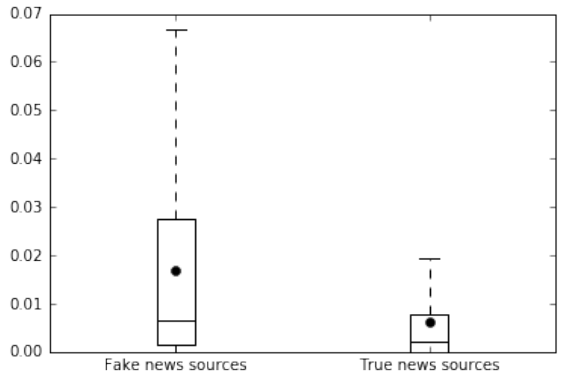
ing the human experts judging about the credibility of the Websites.

The remaining of this Chapter is organized as follows. Section 4.2 presents details of the DistrustRank algorithm. Section 5.4 describes the design of the experiments, and Section 5.5 discusses the results. Section 5.6 discusses previous works on fake News detection. Section 5.7 summarizes the conclusions and presents future research directions.

4.2 DistrustRank Algorithm

To spot unreliable News Websites, without a large annotated corpus, we rely on an important empirical observation: fake News pages are similar to each other. This notion is fairly intuitive, while News Websites approach a broad scope of subjects, unreliable pages are built to mislead people in specific areas, such as fake News about companies, politicians, and celebrities. Additionally, some of the News Websites analyzed share copies of the same unreliable News. Figure 4.1 shows the distribution of the similarity between fake and true News Websites. Using a Wilcoxon statistical test (WILCOXON; KATTI; WILCOX, 1970) with a significance level of 0.05, we verified that the similarity between false News Websites is statistically higher to true News Websites.

Figure 4.1: The distribution of the URL similarity between false and true News domains, where * represent the mean.



The intuition behind DistrustRank is that the credibility score of a website can be regarded as the problem of encountering Websites which headlines do not differ much from fake Websites headlines. To solve this problem, this approach relies on the concept of graph centrality to rank Websites according to their estimated centrality.

We propose to represent the relationship between Websites as a graph, in which the vertices represent the website, and the edges are defined in terms of the similarity between pairs of vertices. We define similarity as a function that measures the textual

similarity of the headlines present in the URLs shared by News Websites. The hypothesis is that fake News Websites have a high centrality index since they are similar to many other fake News Websites. The biased centrality index produces a ranking of vertices' importance, which in this approach indicates the distrust of Websites.

Let L be a set of Websites, and $r \in L$ a tuple $\langle d, u \rangle$, where $r.d$ represents the domain of a website and $r.u$ a set of links for their News. DistrustRank builds a graph representation $G = (V, E)$, where $V = R$ and E is a set of edges that connects pairs $\langle u, v \rangle$ where $v, u \in V$, and uses biased PageRank to calculate centrality scores for each vertex.

The main steps of the DistrustRank algorithm are the following: (a) it builds a similarity graph G between pairs of News Websites; (b) the graph is pruned (G') by removing all edges that do not meet a minimum similarity threshold, which is dynamically calculated based on the average similarity between URLs of fake domains; (c) a search that begins at some particular node v will find the entire connected component containing v; (d) using biased PageRank, the centrality scores are calculated and used to construct a ranking.

The pseudo-code of DistrustRank is displayed in Algorithm (4), where G and G' are represented as adjacency matrices W and W'. In the remaining of this section, we detail the similarity function, and the process to obtain the centrality index ranking.

4.2.1 Similarity between Websites

News Websites usually provide a long link to their News articles which contains the headline of the News, and this link is a good summary of the News article content. For instance, Table 4.1 gives two examples of long links to News articles and their headlines. DistrustRank only takes into consideration the terms (i.e., words) extracted from the long links, represented as unigrams weighted by Term *Frequency-Inverse Document Frequency* (TF-IDF) in order to compute the similarity of pairs of Websites. This choice is motivated by performance issues since for a fast and scalable method, we must be able to handle big graphs, and the extraction of features for comparison cannot be time-consuming. Crucially, to use only the links instead of the full articles' content is a good strategy. In this way, DistrustRank can easily be integrated to search engines, as it does not need additional features.

Therefore, we define the similarity between Websites as the cosine similarity of News headlines, represented by their respective TF-IDF vectors, as detailed in Equation

Algorithm 3 - DistrustRank Algorithm (L, S , β): S

- Input: a set of Websites L, a set of unreliable Websites S and β is the base threshold.
- Output: ordered list O containing the their distrust score.

1: *%building a similarity graph*
2: **for** each $u, v \in L$ **do**
3: $W[u,v] \leftarrow sim\_txt(u.u, v.u)$
4: **end for**
5: *%pruning the graph based on mean similarity of S*
6: $\overline{E} \leftarrow mean\_similarity(S)$
7: **for** each $u, v \in L$ **do**
8: **if** $W[u,v] \geq \overline{E} * \beta$ **then**
9: $W'[u,v] \leftarrow 1$
10: **else**
11: $W'[u,v] \leftarrow 0$
12: **end if**
13: **end for**
14: *%computing a biased centrality*
15: $B \leftarrow BiasedPageRank(W', b)$
16: $N \leftarrow \{\}$
17: *%finding components that contain S*
18: **for** each $s \in S$ **do**
19: $Q \leftarrow \{s\}$
20: **while** there is an edge (u,v) where $u \in Q$ and $v \notin Q$ **do**
21: $Q \leftarrow Q \cup \{v\}$
22: **end while**
23: $N \leftarrow N \cup Q \cap s$
24: **end for**
25: *%reordering N according to their centrality*
26: $O \leftarrow sort\_by\_centrality(N, B)$
27: Return O

4.1.

$$f(u,v) = sim\_txt(u,v) \tag{4.1}$$

where $sim\_txt \in [0,1]$ represents the cosine similarity between the *TF-IDF* vectors of two Websites u and v.

4.2.2 Similarity Threshold (β)

Since centrality in this approach is highly dependent on significant similarity, we can disregard Websites links which the similarity scores are below a minimum thresh-

Table 4.1: Reliable News' URLs, their headlines and Extracted Terms extracted.

URL	News Headline	Terms extracted
www.nydailyNews.com/new-york/education/bronx-teacher-sparks-outrage-cruel-slavery-lesson-article-1.3793930	Bronx teacher sparks outrage for using black students in cruel slavery lesson	[new-york, education, bronx, teacher, sparks, outrage, cruel, slavery, lesson]
www.nypost.com/2018/02/01/mom-teams-up-with-daughter-to-fight-girl-on-school-bus/	Mom teams up with daughter to fight girl on school bus	[mom, teams, up, with, daughter, to, fight, girl, on, school, bus]

old. However, setting an appropriate threshold is a tricky problem (WOLOSZYN et al., 2017b). While a high threshold may mistakenly consider as similar Websites that have very little in common, conversely, a low threshold may disregard important links between Websites.

Using Equation 5.1, we prune the graph based on a minimum similarity between Websites. The result is a weighted graph represented by the adjacency matrix W', where $W'(u,v)$ assumes 1 if an edge that connects u and v exists, and 0 otherwise. To tune the results, we employ a base threshold β that varies according to the mean similarity of false News Websites.

$$W'(u,v) = \begin{cases} 1, & f(u,v) \geq \overline{E} * \beta \\ 0, & otherwise \end{cases} \tag{4.2}$$

In Equation 5.1, $f(u,v)$ is the similarity score according to Equation 4.1; $\overline{E}$ is the mean similarity of the News website dataset, and β is the base threshold.

4.2.3 Biased Centrality

While a regular version of the PageRank algorithm computes a static score to each website, a biased version of PageRank (GYÖNGYI; GARCIA-MOLINA; PEDERSEN, 2004) can increase the score of some specific Websites artificially. A vector of scores is employed to assign a non-zero static bias to a special set of Websites. Then the biased PageRank spreads the bias during the iterations to the pages they point. The matrix

equation of Biased PageRank is:

$$r = \alpha * T * r + (1 - \alpha) * b \tag{4.3}$$

where b is the bias vector of non-negative entries summing up to one, r is the final centrality score, T is the transaction matrix and α a decay factor for bias.

DistrustRank employs a bias to the selected set of seeds (false News Websites) which will be spread to their neighborhoods (similar Websites). The intuition behind this approach is that we can reduce the 'distrust' score as we move further and further away from the bad seed Websites.

Once the centrality scores are computed, we perform a breadth-first search (BFS) on a network graph, starting at some particular node $v \in Seeds$, and explore the neighbor nodes first, before moving to the next level neighbors. The centrality index of the neighbors of v is used to compose the final rank.

4.3 Experiment Design

In this section, we detail the experimental setting used to evaluate DistrustRank. We describe the dataset used, the methods employed for comparison and the metric applied for evaluation, as well as details about DistrustRank parameterization.

4.3.1 Datasets

In order to evaluate this approach, we created two different data sets containing reliable and unreliable News extracted from true News Websites and prominent fake News Websites, as follows:

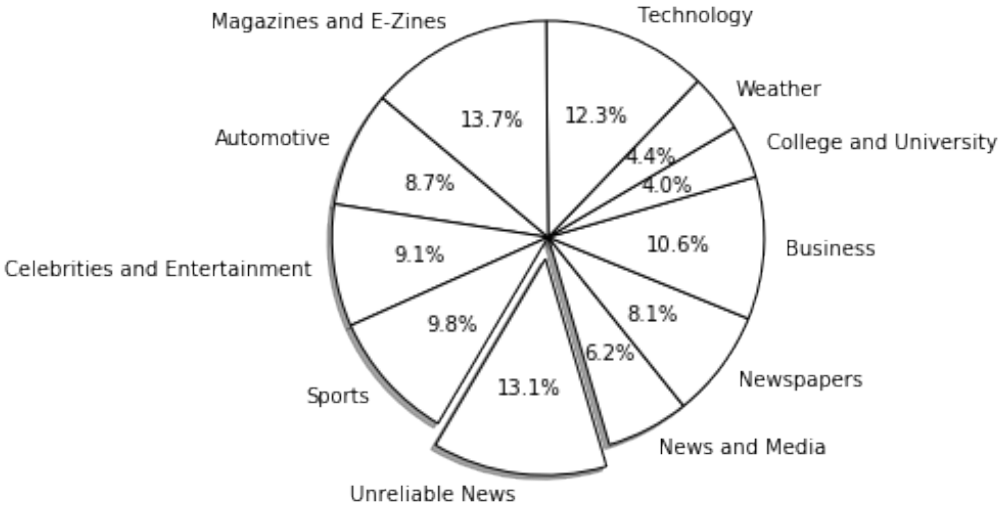
- **Reliable**: we extracted the most popular News Websites from 10 different categories indexed by SimilarWeb[3]. SimilarWeb provides a ranking of the top world News Websites in different categories. The categories used in this set are *Automotive, Celebrities, and Entertainment, Sports, News and Media, Newspapers, Business, College and University, Weather, Technology, Magazines, and E-Zines*. From each of these categories of News, we used the 100 first most popular Websites.

[3]http://www.similarweb.com/top-websites/category/News-and-media

- **Unreliable**: The unreliable News Websites were extracted from the Wikipedia's list of prominent fake News[4]. The total of Websites in this list is 47, which represents nearly 5% of the total of reliable News sources used in this experiment.

For all Websites in both data sets previously listed, we used Internet Archive in order to extract the links to their News articles. Figure 4.2 depicts the distribution of the URLs collected in this task. However, not all of these Websites were employed in the evaluation process, since we are just interested in reliable News that could provide a fair evaluation. For that, we performed a pre-selection of the Reliable News according to the following aspects:

Figure 4.2: Distribution of collected URLs per category of News, where the categories were extracted from http://similarweb.com/



1. We only used reliable News articles that are similar to the unreliable News data set. This choice is motivated to make sure that this approach can identify fake News in a set of similar News since it increases the difficulty of the task and makes it comparable to a real-world problem. Therefore, we compared the intersection of the two sets using the Jaccard similarity coefficient(NIWATTANAKUL et al., 2013). Some News categories, namely *Weather*, *College and University* and *Automotive*, did not achieve a minimum similarity (> 0.4) and therefore were not used in the final data set. Figure 4.5 shows the similarity between Reliable News categories and Unreliable News data set.
2. We only used URLs where the extracted headline contains more than 3 words recognized by an English Dictionary [5]. We only considered headlines extracted from

[4]http://en.wikipedia.org/wiki/List_of_fake_News_websites
[5]http://www.abisource.com/projects/enchant/

the long links, because less than 3 words links do not provide enough information to provide a right classification. Figure 4.4, shows the distribution of URLs per website.

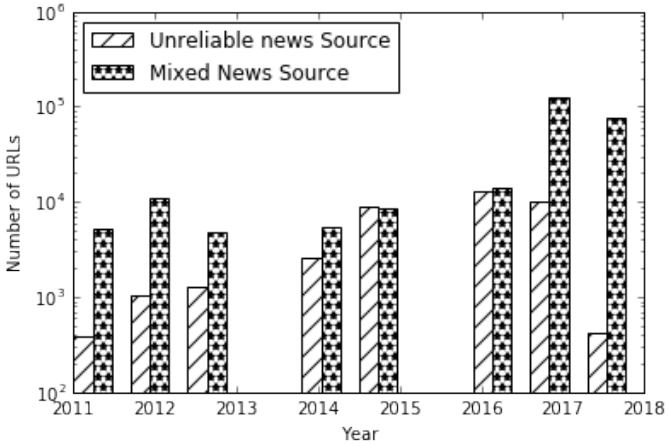
3. we only used News Articles that were published after 2010. This ensures an evaluation that uses a broad scope of News, increasing the diversity of the vocabulary, therefore making the problem harder. Figure 4.3 shows the distribution of the News over the years used in this work.

Table 4.5 provides some statistics about the final dataset employed in this work. From the initial 1000 Reliable News Websites collected, we ended up with 502 following the requirements previously described.

Table 4.2: Summary of the reliable and Unreliable News Websites used in this work.

	Domains	URLs (News)	Terms	URL/Terms
Unreliable	47	37320	158501	4.24
Reliable	502	396422	1281794	3.23

Figure 4.3: Year's distribution of collected News, ranging from 2010 to 2018.



4.3.2 Validation

We adopted k-fold cross-validation, where the unreliable sample is randomly partitioned into k equal size subsamples. For each fold, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times, where at the end of the process all instances in the unreliable set are used for both training and validation, and

Figure 4.4: Distribution of URL's number collected per domain.

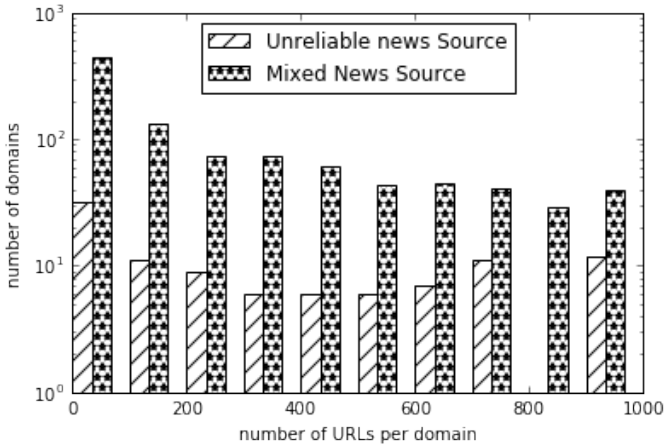
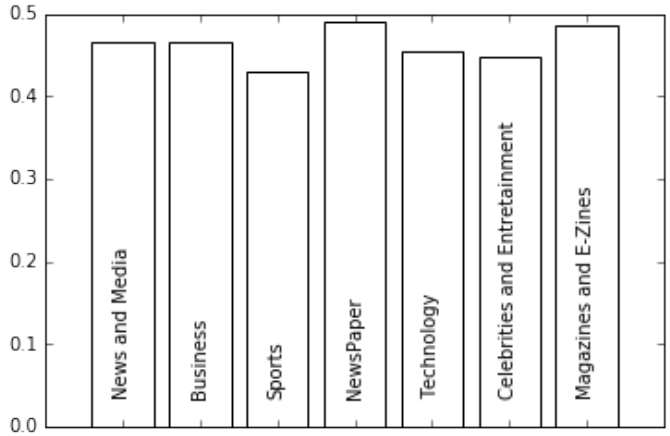


Figure 4.5: Jaccard Similarity between News categories and fake News that achieve the minimum similarity (>0.4).



each observation is used for validation exactly once. The full reliable data set, which contains unlabeled Websites, is used to construct the Graph in all the folds. Finally, the mean precision is computed by using the precision of the k results from each fold, producing a single estimation.

4.3.3 Defining a Similarity Threshold (β)

The parameter β has influenced the results obtained. In order to get more accurate rank, we estimated the best parameter using a numerical optimization method. We used Newton-Conjugate-Gradient, which is employed to minimize functions of multiple variables, where the best β found that minimizes the Mean Squared Error in this dataset is $\beta = 0.849$.

4.3.4 Metrics

In order to evaluate this proposed approach in a classification task we adopted the standard information metrics, such as precision, recall and f1. For the assessment of the ranking task, we used precision@k. The metrics employed ca be briefly described as follows:

- *Precision:* the fraction of the Websites classified as fake that are really fake News. $Precision = \frac{tp}{tp+fp}$
- *Recall* is the fraction of the fake Websites that were successfully identified. $Recall = \frac{tp}{tp+fn}$
- *F-1* corresponds to the harmonic mean between precision and recall. $f1 = 2 * \frac{precision*recall}{precision+recall}$
- *Precision@k* corresponds to the precision using the *k*-firsts elements of the rank.

where tp is the number of positive instances correctly classified as positive, tn number of negative instances correctly classified as negative, fp negative instances wrongly classified as positive, and fn is the number of positive instances wrongly classified as negative. We defined positive instances as fake News Websites and negative instance as reliable News Websites.

4.3.5 Baseline

To measure the gap between this method and a supervised one, we compared the results with the ones using Support Vector Machine, referred to as *SVM*. We employed a linear kernel, recommended for text classification, and which generally uses TF-IDF vectors with many features.

4.4 Results and Discussion

In this section, we present the results and discuss the evaluation of this proposed approach in two different tasks: Ranking of Websites and Binary classification.

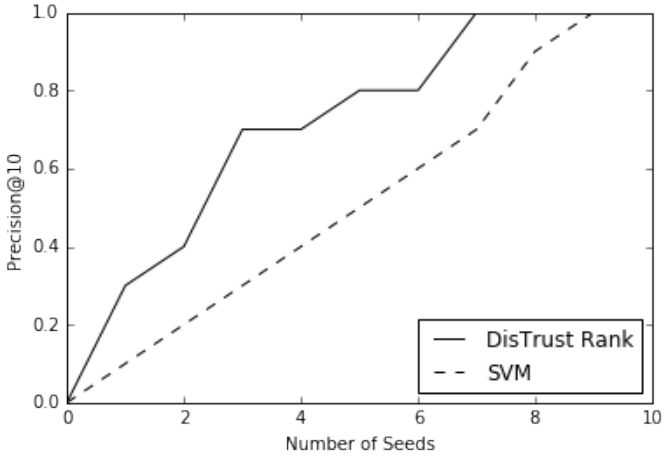
4.4.1 Ranking Task Assessment

To perform a comparison between the ranking of Websites generated by DistrustRank and SVM, we used Precision@10, i.e., we evaluate the precision of the models using the top-10 firsts elements of the rank. We adopted Precision@10 because it usually corresponds to the number of relevant results on the first page of a search engine (e.g., google.com). Additionally, in order to better understand the behavior of the models in a small training data set, we vary from 0 to 10 the number of Websites. It is important to note that for the training step, each model received precisely the same seed set (that was randomly selected from the training set).

Figure 4.6 shows that Distrustrank yields better results for all quantity of seeds analyzed, which are excellent results for a semi-supervised model. While DistrustRank needs only 7 seeds to achieve a precision of 100% (i.e., all the top 10 Websites ranked are genuinely fake), SVM needs 9 seeds to obtain the same precision. Additionally, all the results obtained by this approach using different quantities of seeds showed to be superior to the baseline, where the difference ranges from 10 to 40 percentage points (pp). Using a Wilcoxon statistical test (WILCOXON; KATTI; WILCOX, 1970) with a significance level of 0.05, we verified that DistrustRank results are statistically superior in this task.

As a matter of fact, the good performance of Distrustrank in this task is expected. Supervised learning strategies generally need a large training dataset to yield models with higher predictive power that can generalize well to a new data set. DistrustRank however, is designed considering the empirical observation, that fake News Websites are similar to each other. The use of this domain knowledge in this model, trough a biased graph centrality, allows better performance in small data sets.

Figure 4.6: Number of seeds used to train the model.



4.4.2 Classification Task Assessment

This task consists in predicting the class of a website (e.g., Reliable or Unreliable News website). In this experiment, the positive class represents the Unreliable News Websites and, the Negative class represents Reliable News Websites. We used 2-fold cross-validation, where we randomly shuffle the data set into two sets d_0 and d_1 with equal size. We then train on d_0 and validate on d_1, followed by training on d_1 and validating on d_0. This choice is motivated by the lack of positives instances of fake News Websites.

DistrustRank was initially designed to rank Websites, however, in order to provide a proper evaluation against SVM, we adapted the ranking to act as a classifier. In a classification task, we can compare this approach using complex metrics, such as ratios of false positive and negatives, true positives and negative, as well as precision, recall, and f-1. To transform a ranking into a classification, we used the first k-top Websites of DistrustRank's rank as positive class and the rest of the rank as negative. Setting an optimal value of k without a priori knowledge of the distribution of fake news Websites is a tricky problem. Nonetheless, for evaluation purposes, we use k=47, since we know a priori that this is the number of fake Websites in this data set.

Table 4.3 and 4.4 show that DistrustRank presented a lower error rate in both positive and negative classes, where the differences range from 38 to 47 pp. Additionally, we also analyzed the performance of the models using standard Information Retrieval metrics. Table 4.5 shows that DistrustRank outperformed the SVM model in Precision, Recall and f1, where differences are 16.96, 14.89 and 15.89 pp, respectively.

Table 4.3: Confusion Matrix of DistrustRank.

	Predicted Positive	Predicted Negative
Actual Positive	**36**	11
Actual Negative	9	**493**

Table 4.4: Confusion Matrix of SVM.

	Predicted Positive	Predicted Negative
Actual Positive	**29**	18
Actual Negative	17	**485**

The SVM model presented a similar error distribution among positive and nega-

Table 4.5: Summary of Results

	Precision	Recall	F1
DistrustRank	**0.8**	**0.7659**	**0.7825**
SVM	0.6304	0.6170	0.6236

tive. This was expected since for the learning step we used an equal quantity of positive and negative instances, and that it generally leverages in a learning of an equal distribution of the classes. However, even using a more substantial amount of data for the training, it still presents lower precision and recalls when compared to the proposed approach. In this experiments, we observed that the vocabulary employed by fake News is similar to the one used in reliable News. This textual similarity explains the worst results of the supervised learning model. On the other hand, DistrustRank presented better results using the same amount of data to the training step, due to its semi-supervised strategy.

4.5 Related Work

Several studies have addressed the task of assessing the credibility of a claim. For instance Popat et al. (2016) proposed a new approach to identify the **credibility** of a claim in a text. For a certain claim, it retrieves the corresponding articles from News or social media and feeds those into a distantly supervised classifier for assessing their credibility. Experiments with claims from the website *http://snopes.com/* and from popular cases of Wikipedia hoaxes demonstrate the viability of Popat et al. proposed methods. Another example is TrustRank (GYÖNGYI; GARCIA-MOLINA; PEDERSEN, 2004). This work presents a semi-supervised approach to separate reputable good pages from spam. To discover good pages, it relies on an observation that good pages seldom point to bad ones, i.e., people creating good pages have little reason to point to bad pages. Finally, it employs a biased PageRank using this empirical observation to discover other pages that are likely to be good.

Controversial subjects can also be indicative of dispute or debate involving different opinions about the same subject. Detect and alert users when they are reading a controversial web page is one way to make users aware of the information quality they are consuming. One example of **controversy** detection is (DORI-HACOHEN; ALLAN, 2013) which relies on supervised k-nearest-neighbor classification that maps a Webpage into a set of neighboring controversial articles extracted from Wikipedia. In this approach,

a page adjacent to controversial pages is likely to be controversial itself. Another work in this sense is (PAUL; ZHAI; GIRJU, 2010) which aims to generate contrastive summaries of different viewpoints in opinionated texts. It proposes a Comparative LexRank that relies on random walk formulation to give a score to a sentence based on their difference to others sentences.

Factuality Assessment is another way to asses the information quality. Yu et al.'s work (YU; HATZIVASSILOGLOU, 2003) aims to separate opinions from facts, at both the document and sentence level. It uses a Bayesian classifier for discriminating between documents with a preponderance of opinions, such as editorials from regular News stories. The main goal of this approach is to classify a document/sentence in factual or opinionated text from the perspective of the author. The evaluation of the proposed system reported promising results in both document and sentence levels. Other work on the same line is (RAJKUMAR et al., 2014), which proposes a two-stage framework to extract opinionated sentences from News articles. In the first stage, a supervised learning model gives a score to each sentence based on the probability of the sentence to be opinionated. In the second stage, it uses these probabilities within the HITS schema to treat the opinionated sentences as Hubs, and the facts around these opinions are treated as the Authorities. The proposed method extracts opinions, grouping them with supporting facts as well as other supporting opinions.

There also some works that analyze how a piece of information flows through the internet. For instance, (ECHEVERRIA; ZHOU, 2017) presents an interesting analysis of how Twitter bots can send spam tweets, manipulate public opinion and use them for online fraud. It reports the discovery of the 'Star Wars' botnet on Twitter, which consists of more than 350,000 bots tweeting random quotations exclusively from Star Wars novels. It analyzes and reveals rich details on how the botnet is designed and gives insights on how to detect **virality** in Tweeter.

Other works analyze the writing style in order to detect a false claim. (HORNE; ADALI, 2017) reports that Fake News in most cases are more similar to satire than to real News, leading us to conclude that persuasion in the fake News is achieved through heuristics rather than the strength of arguments. It shows that the overall title structure and the use of proper nouns in titles are very significant in differentiating fake from real. It gives an idea that fake News is targeted for audiences who are not likely to read beyond titles and that they aim at creating mental associations between entities and claims. Decrease the **readability** of texts is also another way to overshadow false claims on the

internet. Many automatic methods to evaluate the readability of texts have been proposed. For instance, Coh-Metrix (GRAESSER; MCNAMARA; KULIKOWICH, 2011), which is a computational tool that measures cohesion, discourse, and text difficulty.

Most of the works just cited rely on supervised learning strategies addressed to assess News articles using few different aspects, such as credibility, controversy, factuality and virality of information. Nonetheless, a common drawback of supervised learning approaches is that the quality of the results is heavily influenced by the availability of a large, domain-dependent annotated corpus to train the model. Unsupervised and semi-supervised learning techniques, on the other hand, are attractive because they do not imply the cost of corpus annotation. In short, this proposed method uses a semi-supervised strategy where only a small set of unreliable News Websites is used to spot another bad News Websites using a biased PageRank.

4.6 Final remarks

This Chapter puts forward a novel semi-supervised approach to spot fake News websites, namely DistrustRank. From a small set of fake News Websites, it creates a graph where vertices correspond to sites and edges to the similarity between the news they share. Next, it applies a biased version of Pagerank(GYÖNGYI; GARCIA-MOLINA; PEDERSEN, 2004) to identify other fake Websites. The similarity is defined regarding the cosine difference of TF-IDF vectors of words extracted from the News links, which usually contains the headline.

The evaluation showed that DistrustRank could effectively identify a significant number of unreliable News (fake News) Websites with fewer data to the training step. In a search engine, DistrustRank can be used either to filter the pages retrieved to the user, or in combination with other metrics to rank search results. The main contributions of this work are the following:

1. a new semi-supervised method to identify Unreliable News Websites, i.e., it does not depend on a large annotated training set;
2. formulation of a similarity function that is computational inexpensive since it only relies on links to represent the similarity between Websites;
3. a better performance in the tasks of ranking and classification, using only a small set of unreliable News Websites;

4. creation of pre-selected data set, containing the News category, date and similarity content; this final data set contains News Websites, long links to the News and their headlines.

As future work, different ways to measure the similarity between Websites are considered. One possible way is using Word Embedding (MIKOLOV et al., 2013), which provides a vector representation that allows words with similar meaning to have similar representation. For instance, this representation could be applied to News links that contain different terms but the same semantic meaning: e.g., *killer* and *murderer*. Another research direction would be to employ different features, such as the time of each News as a decay parameter to measure the similarity between nodes.

4.7 Chapter summary

The work here described makes two majors contributions for this Thesis: a) it proposes a new approach for the detection of Fake News, which can be readily employed to detect Fake Reviews since it relies only on textual similarity; b) it demonstrates that unsupervised (and semi-supervised) is also capable of performing difficult tasks, for instance, the detection of Fake News.

5 PERSONALIZATION OF SUMMARIES

This Chapter presents the work addressed to the second (and last) Research Question proposed in this Thesis: How to create a textual summary of reviews which covers the desirable information. For achieving this goal, BEATnIk were employed. For modeling the user interest and testing our assumptions, reviews collected from collaborative review Websites were used, since they provide the necessary information about users.

5.1 Introduction

Automatic Text Summarization (ATS) - or abstracting - techniques have been widely employed in order to systematically digest a large number of documents and generate in-depth abstracts. Despite fifty years of studies in automatic summarization of texts, one of the still persistent shortcomings is that the individual interests of the readers are not considered. Furthermore, the automatic generation of personalized summaries, which meet the individual profile of the readers, is still underexplored and remains an open research problem.

With this problem in mind, this paper puts forward a new possibility for machine-generated summaries: *personalization*. This technology would play an important role in social network by filtering irrelevant comments; highlighting the interesting aspects of books and movies, or generating a personalized lead paragraph for a news article. Naturally, for an automatic generation of personalized abstracts, information about users interests is necessary. However, for the best of our knowledge, a suitable gold standard to compare the machine-generated summaries with the interest of the reader does not exist yet. To overcome this limitation, we employed an semi-supervised learning strategy, since it does not rely on a large manually annotated training set. Additionally, semi-supervised learning techniques have shown to have a comparable performance in related problems in comparison to supervised models (WU; XU; LI, 2011; WOLOSZYN et al., 2017b).

In this paper, we propose InterestRanking, a semi-supervised algorithm to generate tailored summaries. To accomplish such objective, our approach relies on a mutated version of Google TrustRank (GYÖNGYI; GARCIA-MOLINA; PEDERSEN, 2004). TrustRank is a network link analysis technique first used for semi-automatic separation of useful webpages from spam via a ranking schema. It uses a small set of seed pages, normally evaluated by experts, and use the link structure of the websites to discover, based on their

connectivity, other pages that are likely to be good. The closest a site is to spam resources, the more likely it is to be spam as well. Likewise, in text summarization, ranking schemes are also utilized to find relevant information on documents. Our hypothesis is that a small set of passages that the user demonstrated interest can be used to identify other interests. Correspondingly, the closest - in a graph - a textual passage is to what a user demonstrated interest in, more likely it will be a interest of the reader as well.

InterestRanking constructs a weighted graph where nodes represent passages extracted from the documents to be summarized, connected by edges based on a minimum similarity threshold. The readers' interests are then added into the graph - called seeds -, and then the link structure of the graph is used to discover other passages that are likely to be interesting. While a regular Centrality measures - e.g PageRank - computes a static score to each node, TrustRank can increase artificially the score of specifics nodes. InterestRanking employs a bias to the selected set of seeds - which represents the user interests - which will be spread to their nearest neighbors in the graph. The biased centrality index produces a ranking of node' importance, which in our approach indicates the interest score of a particular passage for a reader. The resulting graph is composed of several components, where each component represents sentences with similar characteristics. Next, a search that begins at some particular seed node v will find the entire connected component containing v. Finally, the centrality index of the neighbors of v are used to compose the final summary.

Our experiments reveal that Interest Ranking significantly outperforms the chosen unsupervised baselines, both in terms of prediction the reader interest, and run-time performance.

The remaining of this paper is organized as follows. Section 5.2 present a scenario where this approach could be applied. Section 5.3 presents details of the 5.6 algorithm. Section 5.4 describes the design of our experiments, and Section 5.5 discusses the results. Section 5.6 discusses previous works on ranking and personalization. Section 5.7 summarizes our conclusions and presents future research directions.

5.2 Use Case

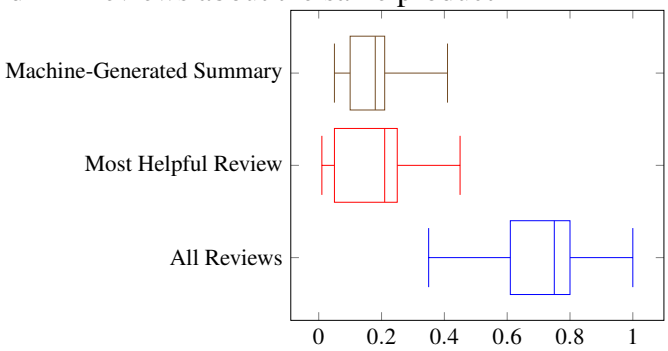
As mentioned before, there is not a suitable gold standard for validation of personalized summaries. Therefore, we evaluate our approach using reviews extracted from Collaborative review websites - e.g. Amazon.com and goodreads.com -, since they con-

tain a rich source of information which include preferences, choices and actions of users - i.e., what a particular user likes and dislikes. The information generated by those websites has been widely employed in other fields, and plays an important role in Recommender System for generation of personalized recommendations.

We have found that the current techniques in this field are not efficient in provide the users a useful short textual summary of products. For instance, those websites commonly provide a textual summary containing the most popular aspects highlighted by the users. Nonetheless, our experiments have shown that - see Figure 5.2 - in average, it only covers around 19% of the the user interests. Another atempt to provide useful descriptions of products is ranking schemes of reviews, using criteria such as helpfulness, usually based on votes given by users. Users are invited to give their feedback about the relevance of reviews using straightforward questions such as “Was this review helpful to you?"; the most voted one becomes ’The Most Helpful Review’, which are usually featured prominently on the website. However, we have found that in average, the ’The Most Helpful Review’ only covers 20.3% of the user’s review about the same product.

On the other hand, we have also found that around 75% of the user’s review can be found in the set of the other reviews about the same product. Such evidence support our claim that it is possible to generate personalized abstracts on Collaborative review websites, since most of the content that the reader is looking for can be normally found in the set of review.

Figure 5.1: Similarity using Jaccard Index of the user’s review with a Summary, The Most Helpful Review, and All Reviews about the same product



5.3 InterestRanking Algorithm

In this work, we propose an unsupervised algorithm to generate personalized summaries based on the concept of *biased centrality*. The intuition behind this strategy is that

the interest score of a passage can be regarded as the problem of detecting passages that do not differ much from previous interest of the user. To solve this problem, our approach relies on the concept of graph centrality giving score to sentences according to their estimated biased centrality.

5.3.1 Text model

We represent the relationship between the passages of the documents to be summarized as a graph, in which the vertices represent the sentence, and the edges are defined in terms of the similarity between pairs of sentences. Formally, let D be a set of documents, and $d \in D$ a set $\{s_1, s_2, ..., s_{|d_i|}\}$, where $|d_i|$ is the number of sentences of the document d_i. InterestRanking builds a graph representation , where $V = \bigcup_{i=1}^{\infty} D_i$ and E is a set of edges that connects pairs $\langle u, v \rangle$ where $v, u \in V$, and represents the similarity between the sentences.

We define the similarity between sentences as the cosine similarity of nodes, represented by their respective *Frequency-Inverse Document Frequency* (TF-IDF) vectors, denoted by $f(u,v) \in [0,1]$. This choice is motivated by performance issues, since for a fast and scalable method, we must be able to handle big graphs and the extraction of features for comparison cannot be time-consuming.

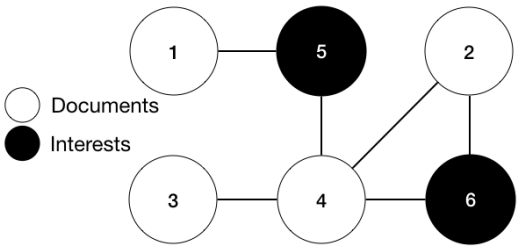
We introduce the transition matrix representations of a textual graph, which will have important roles in the following sections:

$$T = \begin{bmatrix} 1 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 1 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 1 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0 & 0.01 & 1 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0 & 0.01 & 1 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0 & 0.01 & 1 \end{bmatrix}$$

5.3.2 Similarity Threshold (β)

We also remove edges that are below a minimum threshold. However, while a high threshold may mistakenly consider as similar sentences that have very little in common,

Figure 5.2: A simple text graph.



conversely, a low threshold may disregard important links between sentences. Using Equation 5.1, the result is a weighted graph represented by the adjacency matrix W', where $W'(u,v)$ assumes 1 if an edge that connects u and v exists, and 0 otherwise. To tune our results, we employ a base threshold β that varies according to the mean similarity of the textual passages from the documents.

$$T''(u,v) = \begin{cases} 1, & f(u,v) \geq \overline{E} * \beta \\ 0, & otherwise \end{cases} \tag{5.1}$$

In Equation 5.1, $f(u,v)$ is the cosine similarity of Sentences; $\overline{E}$ is the mean similarity of the documents to be summarized, and β is the base threshold.

The transition matrix corresponding to the graph in Figure 5.3.5 is:

$$T' = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

5.3.3 Interests

The premise underlying centrality is that the importance of a node is measured in terms of both the number and the importance of vertices it is related to. Correspondingly, PageRank is based on a mutual reinforcement between pages: the importance of a certain node influences and is being influenced by the importance of some other node. While a regular version of PageRank algorithm computes a static importance score to each node, a biased version of centrality (GYÖNGYI; GARCIA-MOLINA; PEDERSEN, 2004) can

increase artificially the importance score of some specific nodes. We artificially introduce seeds - the user's interests - in the graph and a non-zero static bias is assigned to those special set of seeds, then InterestRanking spreads this bias during the iterations to the nodes they point to. The closest - in a graph - a textual passage is to a seed, more likely it is to be a possible interest of the reader.

For instance, in Figure 5.3.5, assuming that the seed set b is $\{5,6\}$. Nodes 5 and 6 represents the interests, we get the following normalized non-zero static bias:

$$b = \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

5.3.4 Biased Centrality

InterestRanking employs a bias to the selected set of seeds which will be spread to their neighborhoods. The intuition behind this idea is that we can improve the 'interest' score of the passages as we move closer and closer to the seed sentences. The matrix equation of Biased TrustRank is:

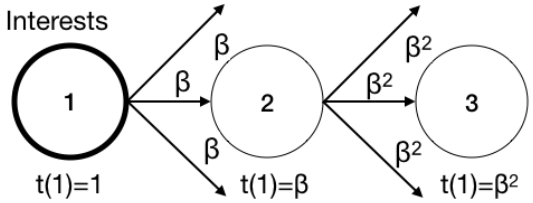
$$r = \alpha * T * r + (1 - \alpha) * b \tag{5.2}$$

where b is the bias vector of non-negative entries summing up to one, r is the final centrality score, T is the transaction matrix and α a decay factor for bias.

5.3.5 Interest Attenuation

As mentioned previously, the interest score increases as we move closer to the interests seeds. Figure 5.3.5 depicts the idea of a **Interest dampening**. Since page 2 is one link away from the interest seed 1, we assign it a dampened trust score of , where $\beta < 1$. Since page 3 is reachable in one step from page 2 with score β, it gets a dampened score of $\beta * \beta$. Considering equation 5.2, in each iteration, the trust score of a node is split among its neighbors and dampened by a factor α.

Figure 5.3: Interest dampening.



5.3.6 Diversity

IN order to ensure diversity, after the computation of the centrality scores, we perform the breadth-first search (BFS) on a network graph, starting at some particular node $v \in Seeds$, and explore the neighbor nodes first, before moving to the next level neighbors. The centrality index of the neighbors of v is used to compose the final rank.

5.3.7 Algorithm

The main steps of the InterestRanking algorithm: (a) it builds a similarity graph G between pairs of sentences; (b) the graph is pruned (G') by removing all edges that do not meet a minimum similarity threshold, dynamically calculated based on the average similarity between all sentences; (c) using biased PageRank, the centrality scores are calculated and used to construct a ranking; (d) a search that begins at some particular node v will find the entire connected component containing v. The pseudo-code of InterestRanking is displayed in Algorithm (4), where G and G' are represented as adjacency matrices W and W'. In the remaining of this section, we detail the similarity function, and the process to obtain the centrality index ranking.

Algorithm 4 - InterestRanking Algorithm (L, S , β): S

- Input: a set of sentences L, a set of sentences S containing the past interest of the user and β is the base threshold.

- Output: ordered list O of sentences containing the potentially interesting for the users.

%building a similarity graph
for each $u, v \in L$ **do**
    $W[u,v] \leftarrow sim\_txt(u.u, v.u)$
end for
%pruning the graph based on mean similarity of S
$\overline{E} \leftarrow mean\_similarity(S)$
for each $u, v \in L$ **do**
    if $W[u,v] \geq \overline{E} * \beta$ **then**
        $W'[u,v] \leftarrow 1$
    else
        $W'[u,v] \leftarrow 0$
    end if
end for
%computing a biased centrality
$B \leftarrow BiasedPageRank(W', b)$
$N \leftarrow \{\}$
%finding components that contain S
for each $s \in S$ **do**
    $Q \leftarrow \{s\}$
    while there is an edge (u,v) where $u \in Q$ and $v \notin Q$ **do**
        $Q \leftarrow Q \cup \{v\}$
    end while
    $N \leftarrow N \cup Q \cap s$
end for
%reordering N according to their centrality
$O \leftarrow sort\_by\_centrality(N, B)$
Return O

5.4 Experimental Design

In this section, we detail the experimental setting used to evaluate our approach. We describe the data set used, the methods employed for comparison and the metric applied for evaluation, as well as details about the parameterization.

5.4.1 Data set

For the validation purposes, we employd data from Amazon.com. The data set (MCAULEY; PANDEY; LESKOVEC, 2015) comprises 19,756 reviews of electronics and 24,234 reviews of books containing identification number (id), numerical rating score and a textual review. In this work, only textual attributes were employed. Regarding the length of the text, only reviews with more then 30 words were employed, since they contain sufficient structural information or cue phrase, which are usually required in ATS based approaches. Table 5.1 describes the profiling of the datasets.

Table 5.1: Profiling of the Amazon dataset.

	Electronics	**Books**
Votes	48.20 ($\pm$ 302.84)	29.71 ($\pm$ 73.58)
Positive	40.12 ($\pm$ 291.99)	20.60 ($\pm$ 64.18)
Negative	8.08 ($\pm$ 22.27)	9.11 ($\pm$ 21.44)
Rating	3.73 ($\pm$ 1.50)	3.41 ($\pm$ 1.54)
Words	350.32 ($\pm$ 402.02)	287.44 ($\pm$ 273.75)
Products	383	461
Total	19,756	24,234

5.4.2 Gold Standard

As discussed before, there still no suitable benchmark to evaluate or approach. To overcome this, we evaluated our summaries in terms of how well they match human-made reviews.

5.4.3 Jaccard Similarity Index

We measure the textual similarity between two documents using Jaccard Similarity Index (NIWATTANAKUL et al., 2013). It is defined as the size of the intersection divided by the size of the union of the sample sets, were "0" means the documents are completely dissimilar, and "1" that they are identical. The corresponding equation is defined as follows:

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{5.3}$$

5.4.4 Baseline

We adopted two different baselines:

- **Gensin Summarizer**[1]: Gensim is a library for Automatic Text Summarization, topic modeling, document indexing and similarity retrieval with large corpora. The summarizing module is based on ranks of text sentences using a variation of the TextRank algorithm (BARRIOS et al., 2016).
- **Most Helpful Review**: usually collaborative reviews Websites provide a mechanism so that users can rate others users' reviews. The most voted review is called the *Most Helpful Review*, and generally, they are displayed on the first page.

5.5 Results and Discussion

In this section, we present the results and discuss the evaluation of our proposed approach using Jaccard Similarity Index (it has a Jaccard Similarity Index value for each review in the review set), where the baselines are referred to as GENSIM (BARRIOS et al., 2016).

The results in Table 5.2 show that our approach outperformed the baselines in all cases. Regarding the mean, the difference in our approach ranges from 14.41 to 17.6 percentage points (pp), with the smallest standard deviation. The table also shows that our results are not only better in average, but also in terms of lower and upper quar-

[1] http://radimrehurek.com/gensim/

tiles, minimum and maximal values, where the differences range from 14.18 to 23.16 pp when compared to the runner-up method, namely GENSIM. The Wilcoxon statistical test (WILCOXON; KATTI; WILCOX, 1970) was applied with a significance level of 0.05, and the result verified that our approach provides statistically superior results.

Table 5.2: Mean Performance using Jaccard Similarity Index, where IR means InterestRanking.

	Q1	Q2	Q3	Mean	std
IR	**0.2326**	**0.2846**	**0.3394**	**0.2865**	**0.067**
Gensin	0.0993	0.1428	0.1936	0.1424	0.083
MHR	0.0867	0.1108	0.1346	0.1105	0.033

5.6 Related Work

Our work relies on existing researches about PageRank. The use of PageRank to generate summaries via ranking schemes have been widely employed by Automatic Text Summarization Systems. For example, LexRank (ERKAN; RADEV, 2004), which relies on the concept of sentence salience to identify the most important sentences in a document. The idea of biasing PageRank to rank documents was introduced in BEATnIk (**??**). It is an unsupervised algorithm for generating biased summaries that cover certain particular aspects. Recent analyses of (biased) PageRank are provided by (**????**) [2, 11]. However, this research is oriented to generate personalized summaries based on previous interests.

5.7 Conclusion and Future Work

In this paper we have put forward a novel semi-supervised approach to generate tailored summaries: InterestRanking. From a small set of interests, it creates a graph where vertices correspond to sentences and edges to the textual similarity between them. Next it applies a biased centrality to rank the passages by interest score. Our experimental results show that we can effectively identify a significant number of interesting passages for the readers with less data to the training step. InterestRanking could be used for different task, for example in social network by filtering/ranking irrelevant comments; highlighting the interesting aspects of books and movies, or generating a personalized

lead paragraph for a news article.

We believe that our work is a first attempt at formalizing the problem and at introducing a comprehensive solution to creation of tailored abstracts. For instance, it would be desirable to further explore the interplay between dampening for interest propagation. In addition, there are a number of ways to refine our methods. For example, instead of selecting the entire seed set at once, one could think of an iterative process: after the oracle has evaluated some nodes, we could reconsider which node it should evaluate next, based on the previous outcome. Such issues are a challenge for future research. Additionally, we would like to consider different ways to measure the similarity between passages, for instance, using Word Embedding (MIKOLOV et al., 2013). Another research direction would be a consolidation of a benchmark for this task.

5.8 Threads to the validity

There is still no standard benchmark for training neither personalized summaries where our model could be tested. To overcome this limitation, we employed reviews extracted from collaborative product's review websites. In such scenarios, the purpose of our model is to mimic the textual revision that a user would write about a particular product.

Our hypothesis is that a summary generated especially for a user who textually covers what would be said by her/him would be much more useful than a general summary. However, in this thesis, we do not evaluate whether this revision that imitates what would be said about a product is more useful than a non-personalized summary. Nevertheless, this does not invalidate our results, since a parameter controls the level of customization, and this can be defined dynamically without the need to change the model. As future work, we consider a qualitative evaluation of the level of personalization in the user opinion.

5.9 Final Remarks

In this chapter, a novel unsupervised approach to generate personalized summaries based on the user's historical data was presented. It creates a complete graph for each item, where each sentence extracted from the Amazon's dataset becomes a node, and a

similarity measure applied between sentences define each edge's weight. Also, it takes into account past reviews from the user (used as a bias) to compute the importance of each sentence. The final summary is based on the centrality score of the sentences weighted by the presence of similar passages from the user.

Our assessment showed that the proposed approach outperformed the baseline Most Helpful Review, as well as an extractive summary of all reviews concerning intersection with the user's reviews.

6 CONCLUSIONS

In this thesis, a new possibility for machine-generated summaries was put forward: personalization. Nevertheless, a distinct benchmark to train and test the proposed hypothesis does not exist yet. To overcome this limitation, we relied on unsupervised and semi-supervised methods since they naturally require no - or less - data for the training step, avoiding the cost of building distinct data sets for this single purpose. Naturally, there are many different suitable unsupervised learning strategies, ranging from those based on closer neighbors to Deep Neural Networks (DNNs). Considering the lack of training data set and the specific hardware for performing the training of these DNNs - which generally require High-Performance Computing, we opted by performing our investigation using graph-based models. Our experiments have shown that unsupervised graph-based models can achieve comparable results in comparison with traditional machine learning techniques, such as Support Vector Machine and computational inexpensive in comparison to the Deep Neural Networks.

To achieve such an overarching end, we divide this wide problem into two sub-research questions, which contributed to the subsequent results and analyses:

- **RQ1 - How to detect a relevant document among a large number of documents?** We introduced a novel unsupervised algorithm called MRR, which is able to identify relevant documents based on the concept of node centrality. In our experiments, we showed that MRR outperformed the prior unsupervised techniques, and has a comparable performance concerning a supervised model. Additionally, it presented a better run-time performance due to a computationally inexpensive textual similarity function. MRR's contributions are the following:

    1. it is an unsupervised method to identify the relevance of documents, i.e., it does not depend on an annotated training set;
    2. centrality scores rely on a similarity function, which needs only two features to represent the similarity between documents, proved to be faster than other graph-centrality methods that are based on documents similarity;
    3. it performs well in different domains (e.g., closed vs. open-ended), as it defines a graph-specific minimum similarity threshold to construct the document graph;
    4. considering documents in two distinct domains, MRR results are significantly

superior to the unsupervised baselines, and comparable to a supervised approach in a specific setting.

- **RQ2 - How to create a textual summary which covers the desirable information for a specific user?** We developed a new unsupervised algorithm based on a biased graph centrality. Our experiments showed that our approach is capable of: a) learning the user's preference and produce an abstract that covers their interests, and b) effectively identify a significant number of unreliable documents with a small training set. The main contributions of this work are the following:

  1. a biased graph-based algorithm to generate personalized summaries that cover the user interest.
  2. a new semi-supervised method to identify Unreliable News Websites, i.e., it does not depend on a large annotated training set;
  3. formulation of a similarity function that is computational inexpensive since it only relies on links to represent the similarity between websites;
  4. a better performance in the tasks of ranking and classification, using only a small set of unreliable News websites;
  5. creation of pre-selected data set, containing the News category, date and similarity content; this final data set contains News websites, along with links to the News and their headlines.

As future work, we would like to consider the use of Deep Neural Networks in our experiments. Once we have a better understanding of the problem, through the research carried out here, we consider the optimization of our models using DNNs. Usually, DNNs achieves better performance; however, it usually requires High-Performance Computing and a more extensive training set. Additionally, we also consider the creation of a unified pipeline for the generation of end-to-end personalized summaries, which integrate all the methods here developed.

REFERENCES

BARRIOS, F. et al. Variations of the similarity function of textrank for automated summarization. **arXiv preprint arXiv:1602.03606**, 2016.

BAYKAN, E.; HENZINGER, M.; WEBER, I. A comprehensive study of techniques for url-based web page language classification. **ACM Transactions on the Web (TWEB)**, ACM, v. 7, n. 1, p. 3, 2013.

CASTRO, M. C.; WERNECK, V.; GOUVEA, N. Ensino de Matemática Através de Algoritmos Utilizando Jogos para Alunos do Ensino Fundamental II. In: . [s.n.], 2016. p. 1039. Disponível em: <http://br-ie.org/pub/index.php/wcbie/article/view/7029>.

CHUA, A. Y.; BANERJEE, S. Helpfulness of user-generated reviews as a function of review sentiment, product type and information quality. **Computers in Human Behavior**, v. 54, p. 547 – 554, 2016. ISSN 0747-5632. Disponível em: <http://www.sciencedirect.com/science/article/pii/S074756321530131X>.

DORI-HACOHEN, S.; ALLAN, J. Detecting controversy on the web. In: ACM. **Proceedings of the 22nd ACM international conference on Conference on information & knowledge management**. [S.l.], 2013. p. 1845–1848.

ECHEVERRIA, J.; ZHOU, S. Discovery, retrieval, and analysis of the'star wars' botnet in twitter. In: ACM. **Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017**. [S.l.], 2017. p. 1–8.

ERKAN, G.; RADEV, D. R. Lexrank: Graph-based lexical centrality as salience in text summarization. **Journal of Artificial Intelligence Research**, v. 22, p. 457–479, 2004.

GANESAN, K.; ZHAI, C.; HAN, J. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 23rd International Conference on Computational Linguistics**. [S.l.], 2010. p. 340–348.

GIRAFFA, L.; MULLER, L.; MORAES, M. C. Ensinado Programação apoiada por um ambiente virtual e exercícios associados a cotidiano dos alunos: compartilhando alternativas e lições aprendidas. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [s.n.], 2015. v. 4, n. 1, p. 1330. ISBN 2316-8889. Disponível em: <http://br-ie.org/pub/index.php/wcbie/article/view/6303>.

GRAESSER, A. C.; MCNAMARA, D. S.; KULIKOWICH, J. M. Coh-metrix: Providing multilevel analyses of text characteristics. **Educational researcher**, Sage Publications Sage CA: Los Angeles, CA, v. 40, n. 5, p. 223–234, 2011.

GUPTA, A. et al. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In: ACM. **Proceedings of the 22nd international conference on World Wide Web**. [S.l.], 2013. p. 729–736.

GYÖNGYI, Z.; GARCIA-MOLINA, H.; PEDERSEN, J. Combating web spam with trustrank. In: VLDB ENDOWMENT. **Proceedings of the Thirtieth international conference on Very large data bases-Volume 30**. [S.l.], 2004. p. 576–587.

HORNE, B. D.; ADALI, S. This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. **arXiv preprint arXiv:1703.09398**, 2017.

HSUEH, P.-Y.; MELVILLE, P.; SINDHWANI, V. Data quality from crowdsourcing: a study of annotation selection criteria. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing**. [S.l.], 2009. p. 27–35.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. **ACM Transactions on Information Systems (TOIS)**, ACM, v. 20, n. 4, p. 422–446, 2002.

KIM, S.-M. et al. Automatically assessing review helpfulness. In: **Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. (EMNLP '06), p. 423–430. ISBN 1-932432-73-6. Disponível em: <http://dl.acm.org/citation.cfm?id=1610075.1610135>.

KLEINBERG, J. M. Authoritative sources in a hyperlinked environment. **Journal of the ACM (JACM)**, ACM, v. 46, n. 5, p. 604–632, 1999.

KUMAR, S.; WEST, R.; LESKOVEC, J. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. **Proceedings of the 25th International Conference on World Wide Web**. [S.l.], 2016. p. 591–602.

LAM, X. N. et al. Addressing cold-start problem in recommendation systems. In: **Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication**. New York, NY, USA: ACM, 2008. (ICUIMC '08), p. 208–211. ISBN 978-1-59593-993-7. Disponível em: <http://doi.acm.org/10.1145/1352793.1352837>.

LI, X. et al. Truth finding on the deep web: Is the problem solved? In: VLDB ENDOWMENT. **Proceedings of the VLDB Endowment**. [S.l.], 2012. v. 6, n. 2, p. 97–108.

LIN, C.-Y. Rouge: A package for automatic evaluation of summaries. In: **Text Summarization Branches Out: Proceedings of the ACL-04 Workshop**. [S.l.: s.n.], 2004. p. 74–81.

MCAULEY, J.; PANDEY, R.; LESKOVEC, J. Inferring networks of substitutable and complementary products. In: ACM. **Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.], 2015. p. 785–794.

MCAULEY, J. J.; LESKOVEC, J. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In: **Proceedings of the 22Nd International Conference on World Wide Web**. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013. (WWW '13), p. 897–908. ISBN 978-1-4503-2035-1. Disponível em: <http://dl.acm.org/citation.cfm?id=2488388.2488466>.

MIHALCEA, R.; TARAU, P. Textrank: Bringing order into texts. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. [S.l.], 2004.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119.

MUDAMBI, S. M.; SCHUFF, D. What makes a helpful review? a study of customer reviews on amazon. com. **MIS quarterly**, v. 34, n. 1, p. 185–200, 2010.

MUKHERJEE, A.; LIU, B. Modeling review comments. In: **Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Jeju Island, Korea: Association for Computational Linguistics, 2012. p. 320–329. Disponível em: <http://www.aclweb.org/anthology/P12-1034>.

NIWATTANAKUL, S. et al. Using of jaccard coefficient for keywords similarity. In: **Proceedings of the International MultiConference of Engineers and Computer Scientists**. [S.l.: s.n.], 2013. v. 1, n. 6.

OLIVEIRA, M. V.; RODRIGUES, L. C.; QUEIROGA, A. Material didático lúdico: uso da ferramenta Scratch para auxílio no aprendizado de lógica da programação. In: . [s.n.], 2016. p. 359. Disponível em: <http://www.br-ie.org/pub/index.php/wie/article/view/6842>.

PAGE, L. et al. The pagerank citation ranking: bringing order to the web. Stanford InfoLab, 1999.

PAUL, M. J.; ZHAI, C.; GIRJU, R. Summarizing contrastive viewpoints in opinionated text. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing**. [S.l.], 2010. p. 66–76.

POIBEAU, T. et al. **Multi-source, Multilingual Information Extraction and Summarization**. [S.l.]: Springer Science & Business Media, 2012.

POPAT, K. et al. Credibility assessment of textual claims on the web. In: ACM. **Proceedings of the 25th ACM International on Conference on Information and Knowledge Management**. [S.l.], 2016. p. 2173–2178.

RADEV, D. et al. Mead-a platform for multidocument multilingual text summarization. 2004.

RAJKUMAR, P. et al. A novel two-stage framework for extracting opinionated sentences from news articles. In: **Proceedings of TextGraphs-9: the workshop on Graph-based Methods for Natural Language Processing**. [S.l.: s.n.], 2014. p. 25–33.

ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: **Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks**. Valletta, Malta: ELRA, 2010. p. 45–50. <http://is.muni.cz/publication/884893/en>.

SAGGION, H.; POIBEAU, T. Automatic text summarization: Past, present and future. In: **Multi-source, multilingual information extraction and summarization**. [S.l.]: Springer, 2013. p. 3–21.

SHARIFF, S. M.; ZHANG, X.; SANDERSON, M. On the credibility perception of news on twitter: Readers, topics and features. **Computers in Human Behavior**, Elsevier, v. 75, p. 785–796, 2017.

SOUZA, T. et al. Semantic url analytics to support efficient annotation of large scale web archives. In: SPRINGER. **Semanitic Keyword-based Search on Structured Data Sources**. [S.l.], 2015. p. 153–166.

STANOVSKY, G. et al. Integrating deep linguistic features in factuality prediction over unified datasets. In: **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**. [S.l.: s.n.], 2017. v. 2, p. 352–357.

TANG, D.; QIN, B.; LIU, T. Learning semantic representations of users and products for document level sentiment classification. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. Beijing, China: Association for Computational Linguistics, 2015. p. 1014–1023. Disponível em: <http://www.aclweb.org/anthology/P15-1098>.

TSUR, O.; RAPPOPORT, A. Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In: **ICWSM**. [S.l.: s.n.], 2009.

WAN, X. Co-regression for cross-language review rating prediction. In: **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**. Sofia, Bulgaria: Association for Computational Linguistics, 2013. p. 526–531. Disponível em: <http://www.aclweb.org/anthology/P13-2094>.

WEST, D. B. et al. **Introduction to graph theory**. [S.l.]: Prentice hall Upper Saddle River, 2001. v. 2.

WILCOXON, F.; KATTI, S.; WILCOX, R. A. Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. **Selected tables in mathematical statistics**, Markham Publishing Co. Chicago, v. 1, p. 171–259, 1970.

WOLOSZYN et al. Mrr: an unsupervised algorithm to rank reviews by relevance. **IEEE/WIC/ACM International Conference on Web Intelligence**, ACM, 2017. ISSN 978-1-4503-4951-2/17/08.

WOLOSZYN, V. et al. Mrr: an unsupervised algorithm to rank reviews by relevance. In: ACM. **Proceedings of the International Conference on Web Intelligence**. [S.l.], 2017. p. 877–883.

WOLOSZYN, V.; SANTOS, H. D. P. dos; WIVES, L. K. The influence of readability aspects on the user's perception of helpfulness of online reviews. **Revista de Sistemas de Informação da FSMA**, v. 18, 2016. ISSN 1983-5604.

WU, J.; XU, B.; LI, S. An unsupervised approach to rank product reviews. In: IEEE. **Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on**. [S.l.], 2011. v. 3, p. 1769–1772.

XIONG, W.; LITMAN, D. Automatically predicting peer-review helpfulness. In: **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**. Portland, Oregon, USA: Association for Computational Linguistics, 2011. p. 502–507. Disponível em: <http://www.aclweb.org/anthology/P11-2088>.

YANG, Y. et al. Semantic analysis and helpfulness prediction of text for online product reviews. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics**. Beijing, China: Association for Computational Linguistics, 2015. p. 38–44. Disponível em: <http://www.aclweb.org/anthology/P15-2007>.

YU, H.; HATZIVASSILOGLOU, V. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the 2003 conference on Empirical methods in natural language processing**. [S.l.], 2003. p. 129–136.

ZENG, Y.-C.; WU, S.-H. Modeling the helpful opinion mining of online consumer reviews as a classification problem. In: **Proceedings of the IJCNLP 2013 Workshop on NLP for Social Media (SocialNLP)**. Nagoya, Japan: Asian Federation of Natural Language Processing, 2013. p. 29–35. Disponível em: <http://www.aclweb.org/anthology/W13-4205>.