# Weighted One Mode Projection of a Bipartite Graph as a Local Similarity Measure

Rotem Stram[1], Pascal Reuss[1,2], and Klaus-Dieter Althoff[1,2]

[1] Smart Data and Knowledge Services Group, German Research Center for Artificial Intelligence, Kaiserslautern, Germany
{rotem.stram,pascal.reuss,klaus-dieter.althoff}@dfki.de
[2] Institute of Computer Science, Intelligent information Systems Lab, University of Hildesheim, Hildesheim, Germany

**Abstract.** Bipartite graphs are a common structure to model relationships between two populations. Many times a compression of the graph to one population, namely a one mode projection (OMP), is needed in order to gain insight into one of the populations. Since this compression leads to loss of information, several works in the past attempted to quantify the connection quality between the items from the population that is being projected, but have ignored the edge weights in the bipartite graph. This paper presents a novel method to create a weighted OMP (WOMP) by taking edge weights of the bipartite graph into account. The usefulness of the method is then displayed in a case-based reasoning (CBR) environment as a local similarity measure between unordered symbols, in an attempt to solve the long-tail problem of infrequently used but significant symbols of textual CBR. It is shown that our method is superior to other similarity options.

**Keywords:** bipartite graph, one-mode projection, textual case-based reasoning, local similarity, weights, long-tail

## 1 Introduction

Complex network analysis is a field that is currently being vastly researched both under theoretical models and for practical use. The bipartite graph is a special type of network where nodes belong to two distinct populations, and includes only connections between population, but not within them. An example of such a graph can be seen in figure 1(a).

Bipartite graphs can model many real world systems, such as economic networks where countries are connected to the products they export [10], or collaboration networks of scientific coauthoring of papers where each author is connected to the paper they (co)authored [13, 18]. Even human preferences can be modeled and studied using bipartite graphs [14, 29].

Many times the goal of researching this type of networks is to model the relationships between items of only one population based on their connections to the other, for instance the economic relations between countries, or coauthorships.

To this end the network is many times projected onto the population we want to focus on, in a process called one-mode projection (OMP). Here nodes from one population are connected to each other if they share at least one neighbor in the bipartite graph.

Looking at the example of coauthorships, many times authors collaborate on more than one paper, some more than others. If we look at authors $l_1$, $l_2$, and $l_3$, authors $l_1$ and $l_2$ could have coauthored five papers together, while authors $l_1$ and $l_3$ only one. Clearly the relationship between $l_1$ and $l_2$ is different from the relationship between $l_1$ and $l_3$. This information is lost if we disregard the number of neighbors $r_i$ two items share in the bipartite graph.

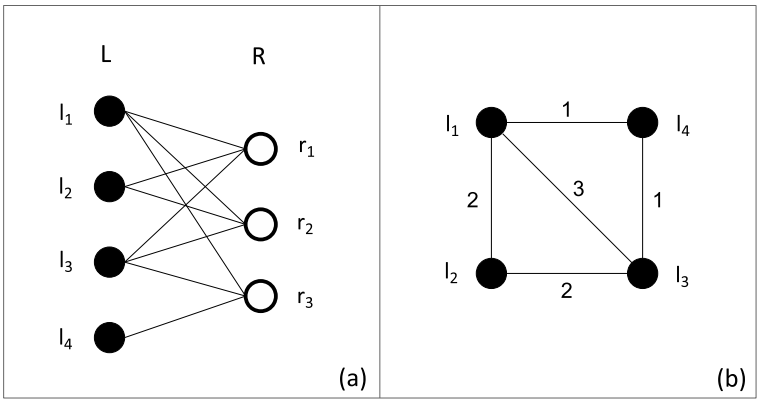


Fig. 1: (a) A bipartite graph consisting of two populations, L and R. (b) The WOMP of the L population with simple weights counting the number of common neighbors

To reduce information loss, several methods have been proposed to take the number of shared neighbors into account and introduce edge weight in the projected network. The simplest form of weighting is to count the number of common neighbors two nodes share [19] (see Fig. 1(b)). When looking at nodes from one population in the bipartite graph, a higher degree may cause a lower impact of the nodes in the other population on each other. As an example we consider again a collaboration network. Two authors who collaborated on a paper might have a stronger connection if they were the sole authors, as opposed to a paper with many other authors. In order to take this additional information into account, Newman introduced a factor of $1/(n_k - 1)$ to each weight, where $n_k$ is the number of authors, or the degree, of paper $k$ [16, 17]. Another problem that might arise with such projections is that adding another connection for authors who already collaborated on many papers before should not have the same impact as a new connection between authors who collaborated on only one or two papers in the past. To add a saturation effect, Li et al. suggested using a hyperbolic tangent function [15].

Another method to evaluate the relationship between two nodes $l_1$, $l_2$ from the same population using a OMP is described by Zweig et al. [30]. Here random graph models are used to find the expected occurrence of a connection motif between two nodes $l_1$ and $l_2$, namely $M(l_1, r, l_2)$, where $r$ is a common neighbor in the bipartite graph, and use it to quantify the *interestingness* of this motif. Only the pairs with the highest interestingness are connected in the OMP. Although the resulting one-mode graph does not contain weights, each edge describes a strong relationship.

A problem that all these methods share is that all weights on the OMP, if they exist, are symmetrical. Going back to our collaboration network example, a new author with very little published papers would likely give a higher weight to his relationship with a new coauthor, than an author who already has many publications. All the methods described 'till now would give the same weight to a connection between these two authors. Moreover, many papers are written by a single author, and this information will be lost in the projection since only collaborations are taken into account. Zhou et al. proposed looking at each connection in a bipartite graph as a resource that is being allocated from nodes of population $L$ to nodes from population $R$, and vice versa [29]. This means that each node $l \in L$ equally distributes its resource to all nodes $r \in R$ it is connected to, and then all nodes $r \in R$ distribute their resources to all nodes $l \in L$ they are connected to. This creates a path between each two nodes $l_1, l_2 \in L$ that share at least one neighbor $r \in R$, with a weight corresponding to the resource allocation between all members of this path. As a result, walking this path from two different directions would result in two different weights.

The methods described above assume that the connections between the populations have an equal weight, and disregard the possibility that the edges in the bipartite graph may be weighted. This work presents a new method to find the weights between two items from the same population that are connected by at least one neighbor in a bipartite graph, while taking into account the edge weights of the bipartite graph, thus creating a weighted OMP (WOMP).

We will first describe our method for WOMP in section 2, then we will discuss its usefulness in modeling similarities between keywords in a textual case-based reasoning (CBR) system in section 3. Experiment results will be shown and analyzed in section 5, demonstrating the superiority of the WOMP over other methods in determining similarities in CBR systems. Section 6 will talk about other works in the CBR field that are related to this work, while the conclusions and future work will be discussed in section 7.

## 2 Method

We turn to look at a bipartite graph with two populations of nodes $L$ and $R$, where each edge between nodes $l_i \in L$ and $r_j \in R$ holds a weight $w_{ij}$. Our goal is to find the weight $w_{ab}^{L \to L}$ between each $l_a, l_b \in L$ that share at least one common neighbor in $R$. To derive this weight we expand the resource allocation method described in [29] to include weights in the original bipartite graph.

The idea behind the resource allocation method is that each node in the graph holds a certain amount of resources, that is then distributed to its neighbors. The weight of an edge then describes part of the resources that is passed along the edge. To find the weight between two nodes from the same population we need to follow the distribution path of the resources.

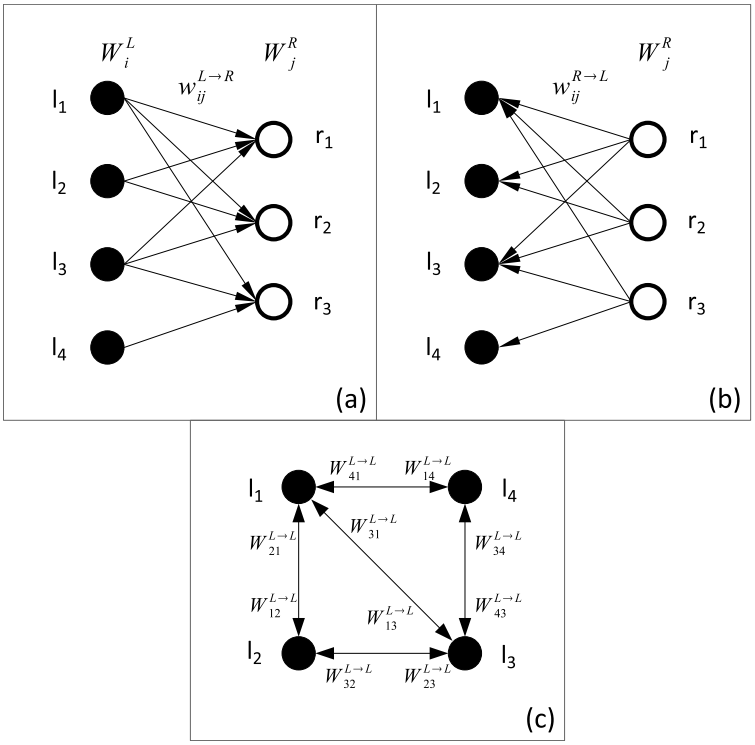


Fig. 2: (a) The flow of resources from population L to R in a bipartite graph. (b) The flow of resources from population R to L in a bipartite graph. (c) The WOMP of the bipartite graph.

Consider a bipartite graph $G(L, R, E)$ where $E$ is the edge list containing tuples $(l_i, r_j, w_{ij})$, where $w_{ij}$ is the weight between nodes $l_i \in L$ and $r_j \in R$, and $|L| = n$, $|R| = m$. Let's say we want to find the WOMP of population $L$. First we define the amount of resources that each node $l_i \in L$ has as:

$$W_i^L = \sum_{j=1}^{m} w_{ij} \tag{1}$$

In case there is no edge between $l_i$ and $r_j$ we consider $w_{ij} = 0$. Next, we define the resource that $l_i$ allocates to $r_j$ as the ratio between the amount of

resources that this edge contributed to $l_i$ and the total amount of resources that $l_i$ possesses:

$$w_{ij}^{L\to R} = \frac{w_{ij}}{W_i^L} \tag{2}$$

It is clear to see that $w_{ij}^{L\to R} \in [0,1]$, and represents the portion of resources that flow through this edge. From here we can conclude that the resources that node $r_j$ accumulates is the sum of those that have been allocated to it from all its neighbors:

$$W_j^R = \sum_{i=1}^{n} w_{ij}^{L\to R} \tag{3}$$

This flow of resources is visualized in figure 2(a). Now we switch directions and distribute resources from $R$ to $L$. Nodes $r_j$ allocate the following to their neighbors $l_i$:

$$w_{ij}^{R\to L} = \frac{w_{ij}^{L\to R}}{W_j^R} \tag{4}$$

The change in flow direction can be seen in figure 2(b). Please note that $w_{ij}^{R\to L}$ is calculated analogously to $w_{ij}^{L\to R}$, as the ratio between the amount of resources that this edge contributed to $r_j$ and the total amount of resources that $r_j$ possesses.

To find the weight between two nodes $l_a, l_b \in L$ one must follow the flow of resources from $l_a$ to $l_b$:

$$w_{ab}^{L\to L} = \sum_{j=1}^{m} p_{aj} \cdot p_{bj} \cdot (w_{aj}^{L\to R} + w_{bj}^{R\to L}) \tag{5}$$

Where $p_{ij} \in \{0,1\}$ indicates whether or not there is an edge between $l_i$ and $r_j$. To make this notion concrete we give an example of how to find the weight $w_{12}^{L\to L}$ between nodes $l_1$ and $l_2$ from figure 1(a). One can see that their shared neighbors are $r_1$ and $r_2$. First we follow the flow of resources from left to right, and then we follow the flow from right to left:

$$w_{12}^{L\to L} = w_{11}^{L\to R} + w_{21}^{R\to L} + w_{12}^{L\to R} + w_{22}^{R\to L}$$

In order to find the weight $w_{21}^{L\to L}$, the same links are used but the flow direction of resources is switched:

$$w_{21}^{L\to L} = w_{21}^{L\to R} + w_{11}^{R\to L} + w_{22}^{L\to R} + w_{12}^{R\to L}$$

One should note that at this stage $w_{ab}^{L\to L} \geq 0$, and allows values greater than 1. To illustrate this we look at another specific case, namely $w_{41}^{L\to L}$, we have:

$$w_{41}^{L\to L} = w_{43}^{L\to R} + w_{13}^{R\to L}$$

It is clear to see that $w_{41}^{L\to L} \geq 1$, since $w_{43}^{L\to R} = \frac{w_{43}}{W_4^L} = \frac{w_{43}}{w_{43}} = 1$ and $w_{13}^{R\to L} \geq 0$. The next step is then to normalize the weights to values in $[0, 1]$, and to do that the following normalization is used:

$$W_{ab}^{L\to L} = \frac{w_{ab}^{L\to L}}{w_{bb}^{L\to L}} \tag{6}$$

Where $w_{bb}^{L\to L}$ describes the highest possible portion of resources that can flow to $l_b$.

This method produces asymmetrical weights for the projection onto population $L$, creating a directed graph where each connection is bi-directional. An illustration of a WOMP can be seen in figure 2(c).

## 3 Similarities in Textual Case-Based Reasoning

In the world of expert systems, an attempt is made to mimic the responses of experts of a given field to certain situations, and possibly to surpass the experts based on some performance measure. Case-based reasoning (CBR) is a paradigm that can be used to implement an expert system. Under CBR, situations may be described in many different ways, from attribute-value pairs, to object-oriented (OO) classes, to graphs.

The idea behind CBR is that similar problems have similar solutions. In order to solve a problem that is described by a situation, an attempt is made to find past situations that are similar to the current one and adapt their solutions to fit the problem [21]. A perfect CBR system would be able to evaluate the a-posteriori utility of each case $c_i$ in the case base to a new problem. This utility function is, however, unknown, and so an approximation attempt is made using heuristics [25]. This means that a CBR system depends heavily on the similarity measure between two situations to perform well.

Two types of similarities are used in CBR, local and global. If we focus on an attribute-value type case description, the local similarity can be defined as the similarity between the values of each attribute. Attributes with numerical values may use a distance measure to define this similarity, while symbolic attributes may utilize taxonomies or similarity tables to model the relationships between the different symbols. The global similarity describes the similarity of whole cases by amalgamating the local similarities. We define $sim_{local}(v, w)$ as the local similarity function between two values $v$, $w$ of a given attribute, and $sim_{global}(c_1, c_2)$ as the global similarity of two cases $c_1$, $c_2$.

Many times the sources for the situation descriptions are in the form of free-text, and a popular method to tackle this is to transform the text into an attribute-value form by extracting wanted features from it [6, 7, 27, 28]. Usually the values are an unordered set of symbols describing keywords and phrases, meaning that the next step is to model the similarity between them. Many times the extracted terms are presented to the experts in the field, and those experts then provide insight into the local similarity. Unfortunately, descriptions in free-text form can cause an explosion of keywords for each attribute, many of

which are informative and descriptive of the situation but are used very rarely. There is only so much information experts can provide a developer about the situation descriptions, and so to best utilize their support experts may be asked to model the relationships only between the most frequently used symbols. This creates a long tail of rarely used attribute values that are informative to the case description, but are excluded from the similarity modeling process. A possible solution to this problem is to simply define $sim_{local}(v, w) = equal(v, w)$ where $equal(v, w)$ is the equality function, if either $v$ or $w$ is unmodeled. This solution is not informative and could affect the quality of the retrieval. In order to prevent this and make full utilization of these values, we propose to use WOMP to supplement our knowledge about the relationships between all values of an attribute.

## 4 Application Area

This work is a contribution to the OMAHA project [1], which is a joint project with Airbus and Lufthansa System to assist aircraft technicians in diagnosing faults using CBR methods. It is a step in the toolchain that was developed in order to tackle the challenges presented by this project [20]. The problem descriptions of past experiences are given in free-text form, and following the toolchain are transformed into an attribute-value form by extracting keywords from the text and assigning them to features. The toolchain was also developed to extract knowledge from the dataset, such as completion rules for the queries, or the importance of each attribute [26]. Although the most common keywords were modeled by the experts in the field, i.e. the experts explicitly quantified their similarity values, many others were disregarded due to time and labor constraints. Our goal is to quantify the similarities of these keywords using WOMP as follows:

1. For each attribute create a bipartite graph where $L$ is the set of all keywords that appear under the given attribute, and $R$ is the set of all possible diagnoses. A keyword $k \in L$ in connected to a diagnosis $d \in R$ if it appeared in a case with diagnosis $d$. The weight of each edge is the number of cases with diagnosis $d$ that $k$ appeared in.
2. Find the WOMP of the keywords $L$ according to the method described in section 2.
3. Use the weights of the edges between the keywords as their similarity value for the given attribute.

Unfortunately the Airbus fault description dataset does not contain well defined diagnoses yet, so in order to test our hypothesis we used a different dataset with similar conditions, namely the internet movie database[1] (IMDb). A casebase was built using the MyCBR tool [5], where each case describes a movie with only one attribute, namely the keywords related to the movie as reported

[1] http://www.imdb.com/

by IMDb, and the diagnosis for each case is the genre of the movie. This means that the system receives a set of keywords as a query, and tries to diagnose the genre by retrieving movies with similar sets of keywords.

## 5 Experimental Results

Two disjoint sets of movie descriptions were constructed by randomly choosing movies that were released between 2005 and 2015, contained a set of keywords, and belonged to one of the following genres: horror, action, romance, and comedy. Short films were ignored. One set contained 6,000 items, namely 1,500 movies from each genre and was used as a training set, while the other contained 500 movies from each genre, 2,000 in total, and was used as the test set.

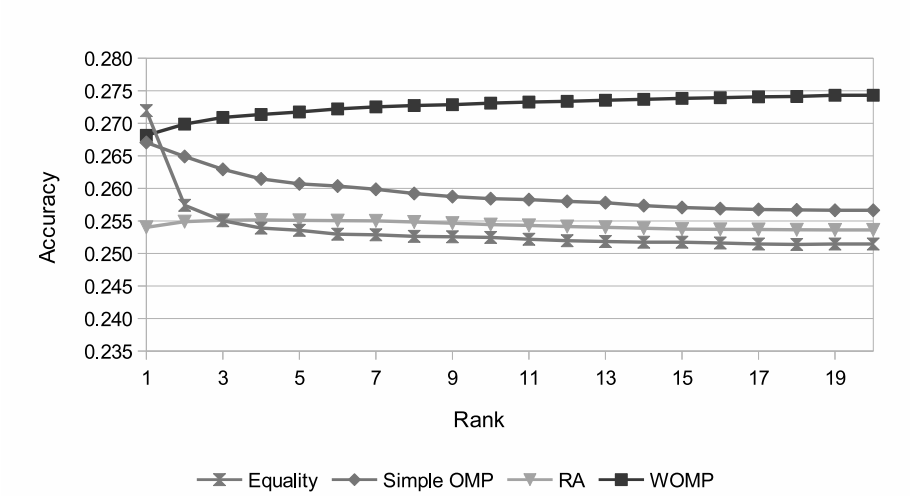


Fig. 3: The accuracy results of the four methods that were tested: the equality function, the simple OMP representing the number of common neighbors, the source allocation (RA) method suggested by Zhou et al., and the WOMP method proposed in this paper.

Four weight functions were used to define the local similarity between the keywords. First, the equality function was used. Then, a bipartite graph was built where population $L$ described the keywords, while $R$ contained the genres. The edge weights described the number of movies each keyword appeared in that belong to the given genre. The second similarity function described the edge weights of the simple OMP, counting the number of neighbors each two keywords shared, disregarding the edge weights in the bipartite graph, and normalizing this number by the maximal degree of the nodes in $L$. The third function was the resource allocation (RA) method described by Zhou et al. [29], where again edge

weights are disregarded. Lastly, the WOMP was used to define the similarity function while utilizing the information in the edges. Only movies from the training set were used to model each similarity. To evaluate how well these similarities performed, four case bases were built from the test set, one for each similarity function, and a retrieval test was performed on each movie in the test set. A case was deemed correctly retrieved if it belonged to the same genre as the query case. To quantify how well each similarity function performed confusion matrices were constructed for the highest ranked retrieved results in the first 1-20 positions. The retrieval accuracy as described by equation 7 was then calculated for each matrix.

$$accuracy = \frac{CorrectDiagnoses}{AllDiagnoses} \tag{7}$$

Figure 3 shows the results of the evaluation. While all four similarity functions performed above the random accuracy level (25%), the WOMP produced the best results. The equality function started off as the best method for the first rank, but then quickly decline and became the worse method starting from the third rank. The graph for the simple OMP is similar in shape to the equality function, however its decline is smoother, and the overall score is higher. The results for the RA method performed the worse, and then the second worse, however its shape is interesting. For the first 5 ranks the accuracy is increased, and then it starts to slowly decline.

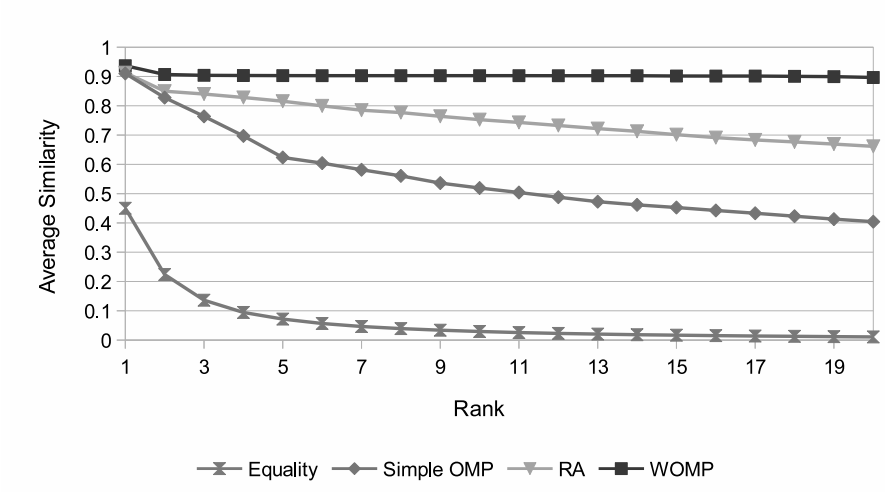


Fig. 4: The average similarity value by rank of the three methods that were tested: the equality function, the simple OMP representing the number of common neighbors, and the WOMP method discussed in this paper.

It is clear that the proposed method, namely WOMP, received the best results starting from rank 2 by a comparably large margin. Another difference that can be seen is the rise in accuracy in the WOMP when considering more ranks, which can be compared to RA, as opposed to the decrease thereof in the other two methods. This can be explained by the differences in average similarity values for different ranks, as shown in Figure 4. The equality function and the simple OMP show a rapid decline for higher ranks. This is probably due to the limited similarity values two keywords can take ($\{0, 1\}$ for the equality function and $\{0, 0.25, 0.5, 0.75, 1\}$ for the simple OMP), and the relatively big step between each value (1 for the equality function and 0.25 for the simple OMP). This leads to low confidence in the lower ranks, and a lower accuracy. The WOMP and RA, on the other hand, produce more finely grained similarity values, creating a continuous and smoother transition between ranks. Even well after rank 10 the average similarity value for WOMP is above 90%, creating a ripe condition for a confidence vote, leading to the rise in accuracy the further we get through the ranks. Even though it is not yet visible at rank 20, logically one can assume that accuracy will decrease after a certain point for WOMP, and the shape of its graph should be comparable to the RA. One can also see that the similarity values for RA are also quite high, even though the accuracy for this method is quite low. This leads to the conclusion that RA may not be a suitable weighting method when the bipartite graph is weighted.

Table 1: The frequency of the keywords for different number of appearances in the dataset with bucket size interval of 100.

| # Appearances | Frequency |
|---|---|
| 100 | 34771 |
| 200 | 191 |
| 300 | 41 |
| 400 | 24 |
| 500 | 8 |
| 600 | 3 |
| 700 | 2 |
| 800 | 2 |
| >800 | 1 |

Table 2: The frequency of the keywords for different number of appearances in the dataset with bucket size interval of 10.

| # Appearances | Frequency |
|---|---|
| 10 | 31930 |
| 20 | 1458 |
| 30 | 559 |
| 40 | 286 |
| 50 | 186 |
| 60 | 133 |
| 70 | 80 |
| 80 | 52 |
| 90 | 45 |
| 100 | 42 |
| >100 | 272 |

In order to demonstrate the long tail abilities of the WOMP, we turn to look at term frequency in the dataset. Table 1 shows the frequency of the keyword under the different buckets of number of appearances with an interval of 100. This table supports the long tail assumption that many keywords are infrequently

used in the dataset. To make matters more precise, table 2 focuses on the first bucket, and divides it into even smaller buckets with an interval of 10. When considering that the dataset contains 35,043 keywords in total, 91% of them appear less than 10 times. Our assumption is that experts do not have the resources to model the similarities of infrequent terms, and in order to demonstrate that WOMP can help with this long tail, we constructed another case base where the 9% most frequent terms remain with the similarity values as calculated by WOMP to simulated the experts' input (since it was shown to produce the best results), and the remaining 91% were modeled with the equality function.

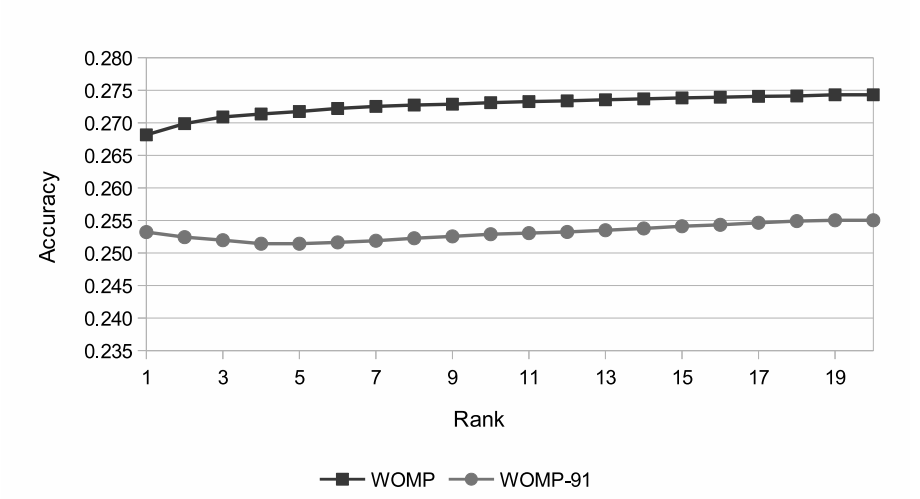


Fig. 5: The accuracy results of the WOMP and a WOMP version where the 91% least frequent terms had their similarity values replaced by those produced by the equality function

Figure 5 compares the retrieval accuracy of the both versions of the WOMP. One can clearly see the boost in accuracy that WOMP provides when it is used to model the similarities of the least frequent terms.

## 6 Related Work

Textual CBR is a well researched field, where problem descriptions and solutions in textual form are processed and transformed into cases that can be compared to each other. Usually, cases are represented in an attribute-value form. One of the first examples of this is PRUDENTIA, a system that transforms legal texts into cases and allows the retrieval of similar cases. This system as described by Weber et al. [27] follows experts guidelines and the strict structure of legal

texts to extract terms and assign them to the correct attribute. The similarity between terms is completely modeled by experts.

A more recent example is described by Bach et al. [4], where cases in attribute-value form were extracted from textual service reports of vehicle problems. Here natural language processing (NLP) methods were used to extract terms from the text. The relationship between these terms was completely modeled by the experts, organizing them in taxonomies and similarity tables. This work is particularly similar to OMAHA, however the long-tail problem of infrequent terms was solved by disregarded anything the experts did not model.

The task of modeling similarities of terms without the help of an expert was tackled by Chakraborti et al. [8]. Here the 1st, 2nd, and 3rd co-occurrence degrees of terms in documents were explored and combined using a weighted sum to find the similarity value between two terms. A similar approach was further explored by Sani et al. [22] who compared the 1st degree co-occurrence with a lexical co-occurrence approach (LCA). In LCA the association *patterns* of two terms are compared in order to produce a similarity value. Both approaches were supplemented by term weights determined by the significance of the term to the domain. These approaches can be seen as a graph problem, although they were not so explicitly defined, and can be compared to the OMP method described by Zweig et al. [30], since significant connections are rewarded. All these approaches, however, disregard the strength of the connection between a term and the document, as connections are unweighted.

There have been several works that explicitely describe the combination of network analysis and graph theory with CBR. Cunningham et al. [9] tackled the textual CBR problem by transforming text into a graph representation by connecting terms according to their sequence of appearance. The similarity measure that was used was maximum common subgraph, meaning that a similarity between individual terms was not necessary. A major drawback of this method is that the complexity of the similarity assessment is polynomial, as opposed to linear when using attribute-value form.

Another work that combines CBR and graph theory is the Text Reasoning Graph (TRG) as described by Sizov et al. [23, 24]. The TRG models causal relationships with textual entailments and paraphrase relations. In their first attempt, the TRG required that the solution of each case contain an analysis part, from which the TRG was extracted. Case similarity was calculated based on the vector space model with TF-IDF weights, while the graph was used only in the reuse step of the CBR cycle [2]. The TRG was later expanded to include the problem description. Two cases are then compared by looking at the problem description part of the graph and finding the so called longest common paraphrase (LCP). Combining the LCP with an informativeness measure of the phrases creates a ranked list of useful cases. As stated before, this method requires an analysis description of how each case was solved, something that may not be readily available in many applications, including our own.

An approach that was similar to ours is described by Jimenes-Diaz et al. [12]. Here OMP was used for link prediction in a recommender system. The idea here was to create a system that recommends programming tasks for students

to practice on, according to previously solved tasks. The authors created a unweighted bipartite graph of tasks and students who solved them and derived the simple OMP, with number of common neighbors in the bipartite graph as weight. These weights were then used as a similarity measure between two tasks with the goal of predicting new links between tasks and users. A comparison was made between several weighting methods, and the simple OMP was found to produce the best predictions. This comparison is closely related to Zhou et al. [29], on which our WOMP method is based, who used resource allocation instead of simple OMP to evaluate the relationships between two nodes from the same population in a bipartite graph. The usefulness of this method was demonstrated on a movie recommendation system, where a user was recommended movies according the the ones he liked in the past. Resource allocation was shown to be a powerful method compared to others.

When looking outside the scope of CBR there have been other attempts at estimating the similarities between object, most notably SimRank [11]. Here a PageRank-like algorithm was utilized to iteratively find similarities between nodes in a graph, with an extension to bipartite graphs. The main idea behind this algorithm is that "two objects are similar if they are related to similar objects." This work was later expanded with SimRank++ to take edge weights into account [3]. Both SimRank and SimRank++, however, produce symmetrical weights and have a relatively high time complexity.

## 7 Conclusions and Future Work

In this paper we presented a novel method to employ edge weights of bipartite graphs when building a OMP of a single population, namely the WOMP. This method is a generalization of the resource allocation based OMP presented by Zhou et al. [29]. The resulting OMP is a directed graph where all edges are bidirectional and are differently weighted in each direction, creating an asymmetrical similarity value between each two nodes in the graph.

This method was then used as a similarity measure between keywords extracted from free text, and evaluated as a supplementary similarity function for textual CBR. The idea here was to use WOMP as a similarity function between keywords that are infrequent but informative and have not been modeled by the experts in the field due to various constraints. An evaluation of the accuracy of WOMP weights, as opposed to the equality function, the simple OMP, and the unweighted resource allocation method was made and it was shown that WOMP produced superior results. A simulation of experts evaluation was also compared to WOMP, and has shown the contribution of this method when weighing infrequent keywords.

The WOMP uses resource allocation to model the relationship between two items from a single population in a bipartite graph. The edge weights of the bipartite graph are regarded as partial resources that make a whole, while each node contains the same amount of resources. This means that weights with different scales but a similar ratio produce the same $w_{ab}^{L \to L}$ values. In the future

we plan on integrating the actual edge weight into the resource, thus allowing different amounts of resources to produce different results even if the scales are the same.

## References


1. German aerospace center - dlr, lufo-projekt omaha gestartet, `http://www.dlr.de/lk/desktopdefault.aspx/tabid-4472/15942_read-45359`
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations and system approaches. AI Communications 7(1), 39–59 (1994)
3. Antonellis, I., Molina, H.G., Chang, C.C.: Simrank++: query rewriting through link analysis of the click graph. In: Proceedings of the VLDB Endowment (2008)
4. Bach, K., Althoff, K.D., Newo, R., Stahl, A.: A case-based reasoning approach for providing machine diagnosis from service reports. In: International Conference on Case-Based Reasoning. Springer Berlin Heidelberg (2011)
5. Bach, K., Sauer, C., Althoff, K.D., Roth-Berghofer, T.: Knowledge modelling with the open source tool mycbr. In: CEUR Workshop Proceedings (2014)
6. Baudin, C., Waterman, S.: From text to cases: Machine aided text categorization for capturing business reengineering cases. In: Proceedings of the AAAI-98 Workshop on Textual Case-Based Reasoning, 51-57 (1998)
7. Brueninghaus, S., Ashley, K.D.: Bootstrapping case base development with annotated case summaries. In: International Conference on Case-Based Reasoning. Springer Berlin Heidelberg (1999)
8. Chakraborti, S., Wiratunga, N., Lothian, Robert andWatt, S.: Acquiring word similarities with higher order association mining. In: International Conference on Case-Based Reasoning. Springer Berlin Heidelberg (2007)
9. Cunningham, C., Weber, R., Proctor, J.M., Fowler, C., Murphy, M.: Investigating graphs in textual case-based reasoning. In: European Conference on Case-Based Reasoning. Springer Berlin Heidelberg (2004)
10. Hidalgo, C.A., Hausmann, R.: The building blocks of economic complexity. In: proceedings of the national academy of sciences 106.26, 10570-10575 (2009)
11. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (2002)
12. Jimenez-Diaz, G., Gómez Martin, P.P., Gómez Martin, M.A., S'anchez-Ruiz, A.A.: Similarity metrics from social network analysis for content recommender systems. In: International Conference on Case-Based Reasoning. Springer International Publishing (2016)
13. Lambiotte, R., Ausloos, M.: N-body decomposition of bipartite author networks. Physical Review E 72.6, 066117 (2005)
14. Lambiotte, R., Ausloos, M.: Uncovering collective listening habits and music genres in bipartite networks. Physical Review E 72.6, 066107 (2005)
15. Li, M., Fan, Y., Chen, J., Gao, L., Di, Z., Wu, J.: Weighted networks of scientific communication: the measurement and topological role of weight. Physica A: Statistical Mechanics and its Applications 350.2 643-656 (2005)
16. Newman, M.E.: Scientific collaboration networks. i. network construction and fundamental results. Physical Review E 64.1, 016131 (2001)
17. Newman, M.E.: Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. Physical Review E 64.1, 016132 (2001)

18. Newman, M.E.: The structure of scientific collaboration networks. In: Proceedings of the National Academy of Sciences 98.2, 404-409 (2001)
19. Ramasco, J.J., Morris, S.A.: Social inertia in collaboration networks. Physical review E 73.1, 016122 (2006)
20. Reuss, P., Stram, R., Juckenack, C., Althoff, K.D., Henkel, W., Fischer, D.: Feature-tak - framework for extraction, analysis, and transformation of unstructured textual aircraft knowledge. In: Proceedings of the 25th International Conference on Case-based Reasoning, ICCBR 2016 (2016)
21. Richter, M., Weber, R.: Case-Based Reasoning: A Textbook. Springer Science & Business Media (2013)
22. Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Term similarity and weighting framework for text representation. In: International Conference on Case-Based Reasoning. Springer Berlin Heidelberg (2011)
23. Sizov, G., Öztürk, P., Aamodt, A.: Evidence-driven retrieval in textual cbr: bridging the gap between retrieval and reuse. In: International Conference on Case-Based Reasoning. Springer International Publishing (2015)
24. Sizov, G., Öztürk, P., Štyrák, J.: Acquisition and reuse of reasoning knowledge from textual cases for automated analysis. In: International Conference on Case-Based Reasoning. Springer International Publishing (2014)
25. Stahl, A.: Learning similarity measures: A formal view based on a generalized cbr model. In: International Conference on Case-Based Reasoning, 507-521. Springer Berlin Heidelberg (2005)
26. Stram, R., Reuss, P., Althoff, K.D., Henkel, W., Fischer, D.: Relevance matrix generation using sensitivity analysis in a case-based reasoning environment. In: Proceedings of the 25th International Conference on Case-based Reasoning, ICCBR 2016. Springer Verlag (2016)
27. Weber, R., Martins, A., Barcia, R.: On legal texts and cases. In: Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop. (1998)
28. Yang, C., Orchard, R., Farley, B., Zaluski, M.: Automated case base creation and management. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer Berlin Heidelberg (2003)
29. Zhou, T., Ren, J., Medo, M., Zhang, T.C.: Bipartite network projection and personal recommendation. Physical Review E 76.4, 046115 (2007)
30. Zweig, K.A., Kaufmann, M.: A systematic approach to the one-mode projection of bipartite graphs. Social Network Analysis and Mining 1.3 187-218 (2011)