Exploratory Recommendations Using Wikipedia's Linking Structure

Adrian M. Kentsch, Walter A. Kosters, Peter van der Putten and Frank W. Takes
{AKENTSCH,KOSTERS,PUTTEN,FTAKES}@LIACS.NL
Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands

Keywords: graph theory, information retrieval, recommender systems, semantic relatedness, ranking algorithms

Abstract

This ongoing research addresses the use of page ranking for computing relatedness coefficients between pairs of nodes in a directed graph, based on their edge structure. A novel, hybrid algorithm is proposed for a complete assessment of nodes and their connecting edges, which is then applied to a practical application, namely a recommender system for books, authors, and their respective movie adaptations. Through relatedness, a level of surprise can be added to the recommender. The recommendation is created by exploring and discovering items of interest beyond the scope of books and authors. These items are then used in an explanatory manner to support the resulting recommendation. The chosen knowledge base is Wikipedia, a suitable source for both computing relatedness coefficients and applying them to the specific recommender system for reading material.

1. Introduction

Recently marking ten years since its launch, Wikipedia became a highly valuable and nearly complete online encyclopedia, with over three and a half million articles kept up-to-date by its active community of contributors (Wikipedia, 2011b). Also, in recent years, it became widely acknowledged as a fair source for reliable information. Beyond its rich content, Wikipedia is furthermore a well-structured collection of webpages representing article-entries that are created and updated to fit certain rules. Its link-structure, namely how articles within Wikipedia link to each other, is the main focus of this research. The motivation for choosing Wikipedia is given by its high level of information and reliability, as well as the particular rules involving the utility of inter-article links (Wikipedia, 2011c). As a consequence to these rules, links can be regarded as article keywords. Although not all links necessarily represent keywords, all words regarded as keywords will also have corresponding links. We assume, supported by other research (Milne & Witten, 2008), that keywords should have a high relatedness score with the article they represent.

Unlike other recommender systems (Melville & Sindhwani, 2010), we propose using Wikipedia's link structure as means to recommend related items to users. This approach is driven by the advantage to have an accurate cross-domain relatedness relationship between items. In other words, items for which relatedness is computed and recommendation is made should not necessarily belong to the same category. Furthermore, with our approach there is no cold start (Schein et al., 2002), a problem frequently occurring with other recommender systems for new users and items. In order to illustrate such a recommender, we propose applying it to a specific domain, which can be regarded as a subset of Wikipedia, namely the subset of thriller/crime fiction authors. This subset includes all linked articles up to a second degree link—links of links—among which the thriller/crime novels and existing movie adaptations would be present too. Such a subset is quite large, as it covers most of the strongly-connected Wikipedia articles, which in turn, as (Dolan, 2011) shows, almost cover the entire Wikipedia.

To give an example for recommending fiction authors, Tom Hanks, the actor, could represent to some users a strong enough reason to relate Dan Brown to Stephen King: he played the leading role in movie adaptations of books written by both authors. The same result may be revealed by collaborative filtering methods (Melville & Sindhwani, 2010), but those are not capable to explain to the user why such recommendation

Appearing in *Proceedings of the 20^{th} Machine Learning conference of Belgium and The Netherlands*. Copyright 2011 by the author(s)/owner(s).

takes place, other than claiming it fits similar user behavior. Furthermore, although certain users may fit the same category of preference, they might still have their preference driven by distinct factors. A more personal motivation given to users when deriving a recommendation is therefore valuable.

The choice to compare items with each other through their respective Wikipedia articles is not driven only by finding motivation, but also by the ability to control the *surprise level* of the recommender. We argue that by selecting items scoring lower in relatedness than others, we can induce a "surprise" element, meaning that the user is familiar with such items less than with highly related ones, also making the derived recommendation less common (Onuma et al., 2009). This is what makes relatedness essential to the algorithm: it facilitates the ability to modify the surprise level. A thorough description of the recommender system will be offered in Section 4.

Semantic analysis and a text-based approach may be, in some cases, more suitable for computing relatedness between articles (Gabrilovich & Markovitch, 2007), but it certainly involves a higher level of complexity which, as research has shown (Milne & Witten, 2008), is not performing better in experiments than the simpler and more efficient link-based alternative. There are, however, several drawbacks to currently existing link-based algorithms for computing relatedness between articles, due to lack of completeness, which will also be addressed later.

In this paper, Wikipedia's set of articles and links is formally treated as a directed graph of nodes–articles, and edges–inter-article links. We treat Wikipedia both as a set of webpages and hyperlinks, and a set of documents and terms, often used as such in information retrieval (Manning et al., 2008), depending on what formulas are described.

Our main interest presented with this paper is to derive a desirable method for computing relatedness between articles that we can apply to the surprise level when offering recommendations. For this particular purpose, taking into account the entire linking-structure of articles is important. Similar research for link-based relatedness have been conducted, either directly on Wikipedia (Milne & Witten, 2008), or generally on graphs (Lin et al., 2007). These either only partially take into account linking properties (Milne & Witten, 2008), and are therefore incomplete, or depend on a hierarchical structure of articles and on walking through the entire graph to compute relatedness (Lin et al., 2007), thus being conditional and expensive. We aim for a complete, unconditional and inexpensive algorithm that needs no further information than links of links for computing relatedness.

We use the following notations throughout the paper: For the article-node A, L_A represents the set of all articles to which A links, while B_A represents the set of all articles that link to A. These are also known as *outlinks* and *inlinks* respectively (Langville & Meyer, 2006). However, in Wikipedia the latter is called the set of *backlinks* of A (Wikipedia, 2011a), whereas the former is called the set of *links*. For this paper, we opt for Wikipedia's vocabulary. We also represent the link-edge oriented from A_1 to A_2 by $A_1 \to A_2$ and the set of all Wikipedia articles by W. Furthermore, all formulas that we present here are adapted by us to fit the above notations.

The remainder of this paper is divided into five sections: Section 2 describes the link-structure of Wikipedia's corresponding graph and how relatedness between articles is computed; Section 3 takes a step further towards ranking nodes in a graph and weighting certain relatedness computations more than other; Section 4 presents the practical application of Wikipedia's graph-relatedness to the actual recommender system for thriller/crime fiction; Section 5 illustrates the advantages of such an algorithm through concrete examples; and finally Section 6 presents a conclusion and proposed future work.

2. Link Analysis and Relatedness

There are several scenarios that carry information regarding the way two articles, A_1 and A_2, might relate to each other, as illustrated in Figure 1. One scenario is represented by the *direct link* between the two articles. If one direct link exists, formally when either $A_2 \in L_{A_1}$ or $A_1 \in L_{A_2}$, it intuitively implies the articles are more related than if it does not exist. Another scenario, implying an even stronger relatedness, would be the *reciprocal link*, meaning that both articles directly link to each other. The *directed paths* would represent a third scenario, meaning there is no direct link, but there are one or more directed paths with intermediate nodes. In practice, given that Wikipedia represents a graph of strongly connected articles, directed paths exist for almost every pair of articles, the average shortest path being of length 4, as found by (Dolan, 2011). Thus, we believe that only the directed paths with one node in between significantly contributes to relatedness. This in other words is equivalent to a shortest path, unweighted, of length 2, occurring when $L_{A_1} \cap B_{A_2} \neq \emptyset$ or $L_{A_2} \cap B_{A_1} \neq \emptyset$.

Two more scenarios regarding an intermediate node are represented by *shared links* and *shared backlinks*. They formally occur in the following situations: in the former case when $L_{A_1} \cap L_{A_2} \neq \emptyset$, and in the latter case when $B_{A_1} \cap B_{A_2} \neq \emptyset$. In information retrieval these are also known as *co-reference* and *co-citation* respectively (Langville & Meyer, 2006). Usually, the more shared links or shared backlinks between two articles, the more related the articles are. There are several methods to normalize a relatedness coefficient based on the number of shared articles, the *Jaccard index*, computed by dividing the size of the intersection by the size of the union, being one of them (Lin et al., 2007); but a more complex approach that also relates the result to the size of the entire set of articles is preferred. (Milne & Witten, 2008) proposes two different methods for computing relatedness, one for shared links and one for shared backlinks, both methods being commonly used in information retrieval with text and queries. For shared backlinks, the *Normalized Google Distance* $NGD\left(A_1, A_2\right)$ between articles A_1 and A_2 is defined by (Milne & Witten, 2008) as:

$$NGD\left(A_1, A_2\right) = \frac{\log\left(\max\left(\left|B_{A_1}\right|, \left|B_{A_2}\right|\right)\right) - \log\left(\left|B_{A_1} \cap B_{A_2}\right|\right)}{\log(|W|) - \log\left(\min\left(\left|B_{A_1}\right|, \left|B_{A_2}\right|\right)\right)} \tag{1}$$

where W stands for the set of all Wikipedia articles. Note that the size of each set is taken into account, representing the number of backlinks. A shared backlink by definition implies there is an article containing both terms. The Normalized Google Distance calculates how related terms are by their occurrence separately and together in all other webpages (Cilibrasi & Vitanyi, 2007), which we agree is suitable for our case. The values for this function range from 0 to ∞, so the result needs to be further normalized to range as a coefficient, from 0 to 1, not as distance. (Milne & Witten, 2008) proposes to invert the values between 0 and 1 and ignore all values that fall beyond, which, given that a distance higher than 1 implies *negative correlation* (Cilibrasi & Vitanyi, 2007), is a fair assumption. We define the *relatedness coefficient between A_1 and A_2 through backlinks $RC_B\left(A_1, A_2\right)$* as:

$$RC_B\left(A_1, A_2\right) = \begin{cases} 1 - NGD(A_1, A_2) & 0 \leq NGD(A_1, A_2) \leq 1 \\ 0 & NGD(A_1, A_2) > 1 \end{cases} \tag{2}$$

For relatedness via shared links, (Milne & Witten, 2008) proposes the *cosine similarity of* `tf` × `idf` *weights*, another method popular with information retrieval (Manning et al., 2008) measuring how important terms are to documents containing them, in our case links to articles. Basically, each shared link is first weighted using the `tf` × `idf` formula as follows:

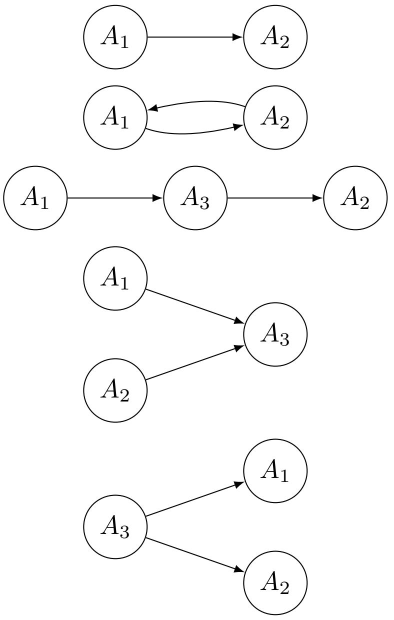


Figure 1. The *five scenarios* for A_1—A_2 relatedness, top–down: the *direct link*, the *reciprocal link*, the *directed path*, the *shared link* and the *shared backlink*.

$$w_{A' \to A} = \left(\mathtt{tf} \times \mathtt{idf}\right)_{A' \to A} = \frac{1}{\left|L_{A'}\right|} \times \log \frac{|W|}{\left|B_A\right|} \tag{3}$$

where A is the *term*–article or link, and w_A is its *weight*. The formula is simplified twice, firstly because the `tf` formula does not need to count the number of times the same link occurs in an article because in Wikipedia this should only happen once if it exists. (Milne & Witten, 2008) further simplifies this formula, completely eliminating `tf` from the equation, after observing it does not affect the final result. Thus, the value of the weight becomes the `idf` of the term–article, independent from the *document*–article:

$$w_A = \left(\mathtt{tf} \times \mathtt{idf}\right)_A = \mathtt{idf}_A = \log \frac{|W|}{\left|B_A\right|} \tag{4}$$

All links are combined and consequently normalized with the following expression form for the *cosine similarity*, representing the *relatedness coefficient between A_1 and A_2 through links $RC_L\left(A_1, A_2\right)$*:

$$RC_L\left(A_1, A_2\right) = \frac{\sum_{A \in L_{A_1} \cap L_{A_2}} {w_A}^2}{\sqrt{\sum_{A \in L_{A_1}} {w_A}^2} \times \sqrt{\sum_{A \in L_{A_2}} {w_A}^2}} \tag{5}$$

We chose to also simplify this equation from its standard form, thus illustrating the summations of weights for existing links only, the weights for non-existing links being 0 and therefore not included above.

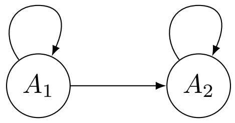


Figure 2. By assuming A_1 and A_2 both link to themselves, a direct link $A_1 \rightarrow A_2$ implies that A_1 becomes a *shared backlink* and A_2 becomes a *shared link* between A_1 and A_2.

(Milne & Witten, 2008) shows that both methods to compute relatedness between articles on Wikipedia perform better in experiments than semantic methods and proposes an arithmetic average of the two relatedness coefficients to be computed for the overall coefficient. We believe this is not always justified, and thus introduce a weighted average $\overline{RC}_{(L,B)}$ instead:

$$\overline{RC}_{(L,B)} = \alpha\, RC_L + (1{-}\alpha)\, RC_B \tag{6}$$

where α is a variable between 0 and 1, depending on which relatedness should be weighted more than the other. Furthermore, (Milne & Witten, 2008) omits the directed or reciprocal link scenarios, which should also be treated. In fact, if such scenarios occur, their formulas decrease the relatedness coefficient instead of increasing it. Our first "fix" is to assume all articles link to themselves too, as illustrated in Figure 2. We keep this assumtion for the remainder of this paper: $A \in L_A$ and $A \in B_A$. A direct link, however, should have a stronger influence on relatedness than just an extra shared link. Therefore, we introduce another variable, β, and define the overall $\overline{RC}$ as:

$$\overline{RC} = \overline{RC}_{(L,B)} + \beta \left(1 - \overline{RC}_{(L,B)}\right) \tag{7}$$

In the next section, we present the derived formulas for variables α and β. The directed paths will also be treated and included in β's formula.

3. Page Ranking and Weighting

When computing relatedness through shared links and shared backlinks, information about each article's "parents", "children" and "other parents of its children" is required, which means backlinks, links and backlinks of links. This is formally known as the article's *Markov blanket.* In this section we will also require the "other children of parents", the links of backlinks, for computing what is known as a particular type of a *Markov chain* used for ranking webpages, namely the *Google PageRank.*

Algorithms that iterate throughout the whole graph are used for webpage ranking, but also for computing relatedness. Though they are expensive, they can discover useful information. Google's PageRank for example computes the probability that a random surfer visits a certain article (Langville & Meyer, 2006), thus determining which pages attract more visitors. We are using a simplified PageRank, iterating only once and only through the subgraph defined by the article itself, its backlinks and the links of its backlinks. The article's links are also included, because of our assumption that all articles link to themselves too, making the computational space an inverse of a Markov blanket. The PageRank PR_A for article A is formally given as:

$$PR_A = \sum_{A' \in B_A} \frac{1}{|L_{A'}|} \tag{8}$$

This measure indicates the chance that someone visiting a backlink of an article eventually arrives on that particular article too. We believe this is a good candidate for variable α from Equation 6 not only because it can determine whether backlinks are more important than links for relatedness, but also because it adds what Equation 2 was ignoring: weighing those backlinks similarly to how Equation 4 weighs links before combining them in Equation 5. With another popular ranking algorithm called *HITS*, this is equivalent to determining whether articles have a higher *authority* score or a higher *hub* score (Kleinberg, 1999).

A further application for PageRank is found for the directed paths, whose influence on relatedness is determined by how much, say, A_1 contributes to the rank of A_2 via the respective paths. The more paths there are, the higher the relatedness will be. Keeping consistency with our chosen computational space, these paths will be restricted to a maximum length of 2, having only one intermediate node, longer paths being ignored. The PageRank can therefore not only determine α, but also β, which can then be used for both the *direct link* and the *directed paths.* As Equation 8 shows, the rank transferred from one page to another via its links is equally distributed, becoming the rank of the page divided by its number of links. With this observation in mind, we can define the PageRank $PR_{A_1 \rightarrow A_2}$ of a path $A_1 \rightarrow A_2$ as follows:

$$PR_{A_1 \rightarrow A_2} = \tfrac{1}{|L_{A_1}|} \left(|\{A_2\} \cap L_{A_1}| + \textstyle\sum_{A \in L_{A_1} \cap B_{A_2}} \tfrac{1}{|L_A|} \right) \tag{9}$$

Earlier we mentioned that similar iterative algorithms exist to directly compute relatedness rather than ranking, *SimRank* (Jeh & Widom, 2002) and *PageSim* (Lin et al., 2007) being two of them. These can be considered related research, but unlike our approach, they are even more expensive than PageRank, computing values for pairs of nodes instead of just nodes. Furthermore, they also require a hierarchically structured graph (Jeh & Widom, 2002), incompatible to Wikipedia's. However, the improvement to SimRank used by (Lin et al., 2007) with the introduction of PageSim is similar to our own improvement to (Milne & Witten, 2008). Concretely, (Lin et al., 2007) also implement a version of the path PageRank and take advantage of the complete linking structure. Other than this, their research has a different scope.

In order to determine α from Equation 6, we need to take into account the PageRanks for A_1 and A_2 representing the articles compared. We do this by taking the arithmetic average of the two articles, but we also need to normalize the result. In theory, using our formula, PageRank can take values ranging from 0 to the number of all articles involved. Since there is only one iteration, the articles involved represent the union of backlinks B_{A_1} with backlinks B_{A_2}. Furthermore, before iteration, all PageRanks are equal to 1. After normalization, these values should become equivalent to a minimum of 0, a maximum of 1 and an initial or neutral value of 0.5. Therefore, the following equations are considered to determine α:

$$\alpha' = \frac{1}{2}\left(PR_{A_1} + PR_{A_2}\right) \tag{10}$$

$$\alpha = \frac{\alpha'\left(|B_{A_1} \cup B_{A_2}| - 1\right)}{\alpha'\left(|B_{A_1} \cup B_{A_2}| - 2\right) + |B_{A_1} \cup B_{A_2}|} \tag{11}$$

where α' is the value before normalization. The PageRank of a path does not require normalization, because it has no neutral value and takes values ranging between 0 and 0.5. Thus, the proposed formula for β, is simply the arithmetic average of the path PageRanks:

$$\beta = \frac{1}{2}\left(PR_{A_1 \to A_2} + PR_{A_2 \to A_1}\right) \tag{12}$$

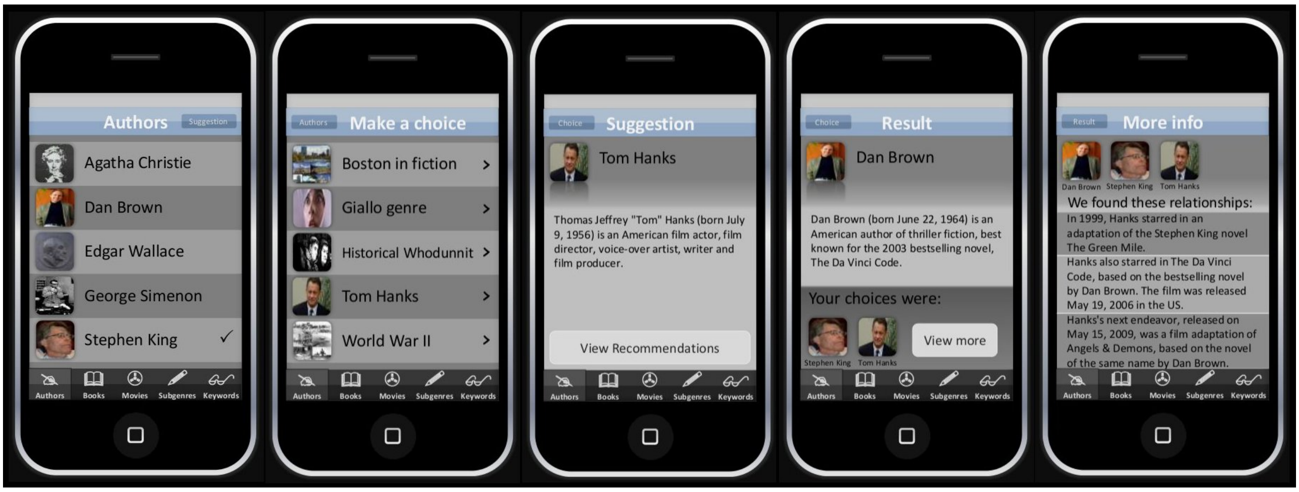
To sum up, given two articles A_1 and A_2, their relatedness coefficient via backlinks is derived in Equation 2, and via links in Equation 5. Then, Equation 6 offers a weighted average formula between the two types of relatedness, whose α is computed using the normalized PageRank. Finally, Equation 7 takes into account the entire linking structure of articles, by considering direct links and directed paths too. This is done through β, which uses the path PageRank. When α is equal to 0.5, it equally weighs relatedness over links with the one over backlinks, and when β is 0, it means there are no direct links and no directed paths.

4. Recommender System

As mentioned in the beginning, the Wikipedia-based graph of articles and their corresponding computed relatedness coefficients are applied to an interactive and exploratory recommender system. This represents a learning mechanism for both the user and the recommender, through which it is discovered what exact items of interest drive the user to the end-resulting recommendation. The domain of thriller/crime fiction authors is chosen to exemplify this recommender as follows: the user is first prompted with a list of authors from which to select the ones that he or she is interested in; next, a list of related items that do not belong to the category of thriller/crime authors is shown, from which the user is again invited to select what is of interest; the iteration can be repeated if still necessary—if more authors could still be suggested—or the user could opt for an immediate recommendation; when this recommendation is shown, it will be complemented by an overview of all selected items and how they relate to each other, to the initially selected thriller/crime authors, and to the resulted recommendation. This is done by simply quoting the paragraphs from which the mentioned articles link to each other.

Because Wikipedia's policy is to only link once from an article to another, searching for paragraphs containing the reference actually implies searching for keywords, not links. Sometimes these keywords are more difficult to find because alternative text can be used for links, also known as anchor text or link label. For example in "Dan Brown is American" the word "American" is an *alternative text* linking to the "United States" article. There are several methods to improve this search, the simplest being to only look for keywords representing the linked article title—"United States"—and its alternative text—"American"—neglecting the rest. A semantic approach is sometimes more suitable (Gabrilovich & Markovitch, 2007), but it falls beyond the scope of our research.

A special feature of our recommender system is the *surprise level*, which can be easily modified by choosing between different ranges of relatedness values. We claim that if the highest related articles are chosen, say the ones with a coefficient above 0.750, the end result will correspond to the prediction of other recommender systems. However, if we limit relatedness values to a range between 0.250 and 0.500, more *surprising* (Onuma et al., 2009) intermediary items will be shown, but the end result can still be justified using the same approach that mentions how all articles relate to each other. In fact, besides filtering out articles that are nearly unrelated—with a coefficient below 0.250—the main use of the relatedness coefficient is to be able to modify this surprise level. If the user is interested in finding the most similar author to his preference, thus aiming for low surprise, then he or she will receive a



Thomas Jeffrey "Tom" Hanks (born July 9, 1956) is an American film actor, film director, voice-over artist, writer and film producer.

Dan Brown (born June 22, 1964) is an American author of thriller fiction, best known for the 2003 bestselling novel, The Da Vinci Code.

In 1999, Hanks starred in an adaptation of the Stephen King novel The Green Mile.

Hanks also starred in The Da Vinci Code, based on the bestselling novel by Dan Brown. The film was released May 19, 2006 in the US.

Hanks's next endeavor, released on May 15, 2009, was a film adaptation of Angels & Demons, based on the novel of the same name by Dan Brown.

Figure 3. Mobile Application screenshots exemplifying the recommendation path from author "Stephen King" to author "Dan Brown" when selecting "Tom Hanks" as keyword.

list with the most related items. If, however, the user aims for a more surprising recommendation, and consequently a more exploratory experience, the surprise level is increased, such that the most commonly known items will be hidden in favor of the least expected ones.

Our approach also presents several other advantages when compared to popular recommender systems such as collaborative filtering. The problem of *cold start* (Schein et al., 2002), for example, which occurs when the system does not hold sufficient information on users and items to reach a recommendation, is not present with our approach because we extract all necessary information from Wikipedia. Also, our algorithm works when no input is selected, when, in the presented case, no fiction author is initially selected by the user. This for example may happen when the user does not yet know any author to be able to select from. Since it is already known what articles highly relate to authors, or what articles are shared among two or more authors, a selected list of articles can always be offered given the user's preference for the surprise level: more expected or more surprising.

All relatedness coefficients between authors and their "neighbors"—links and backlinks—are precomputed and stored in a database as soon as the required information is synchronized with Wikipedia's, guaranteeing a quick response for the recommender system. We call these "neighbors" *keyword-articles* and we precompute their relatedness not only to authors but to pairs of authors too, therefore knowing immediately which of them relate most to each other. We can also precompute related keyword-articles to more than two authors taken together, although for our recommender and for simplicity we prefer to keep them paired: one author from the list chosen by the user, and the other from the list of potential results. After author selection, we display the keyword-articles in the order of relatedness with pairs of authors.

Relatedness with a pair should not only take into account the relatedness computed with each author separately, but also how close to each other these coefficients are. Thus, we define *RF* to be the *relevance factor* and *RC* the *relatedness coefficient between term A and the pair* $\{A_1, A_2\}$ as follows:

$$RF\left(A, \{A_1, A_2\}\right) = \frac{\min(RC(A,A_1),RC(A,A_2))}{\max(RC(A,A_1),RC(A,A_2))} \qquad (13)$$

$$RC(A,\{A_1,A_2\}) = RF(A,\{A_1,A_2\}) \times \frac{RC(A,A_1)+RC(A,A_2)}{2} \qquad (14)$$

Therefore, the higher and closer to each other the two relatedness coefficients are, the higher the combined relatedness is.

Our proposed algorithm for the recommender system, which we are implementing for a website and a mobile application, is as follows: first let the user select one or more fiction authors from the input and choose whether the recommendation should be expected or surprising; then for all selected authors, take all keyword-articles shared with unselected authors, order them by the relatedness coefficient with the respective pair of authors and display the ones fitting the surprise criteria for a new selection; let the user select one keyword-article from the list and reveal the unselected author linked to it as recommendation; if more results fit the surprise criteria, opt for the highest relatedness; for the result, display all articles that have been involved together with all paragraphs in which the ar-

Table 1. Relatedness of "Dan Brown", "Stephen King", and both authors taken as pair, with their shared links and backlinks, regarded as keyword–articles.

Article Title	Dan Brown	Stephen King	both as pair
James Patterson	0.625	0.536	0.497
Tom Hanks	0.501	0.475	0.462
Ron Howard	0.624	0.420	0.351
The Da Vinci Code	0.978	0.427	0.306
Akiva Goldsman	0.558	0.368	0.305
Arthur Machen	0.343	0.561	0.276
Left Behind	0.586	0.339	0.267
Roger Ebert	0.224	0.398	0.175
New England	0.167	0.252	0.138

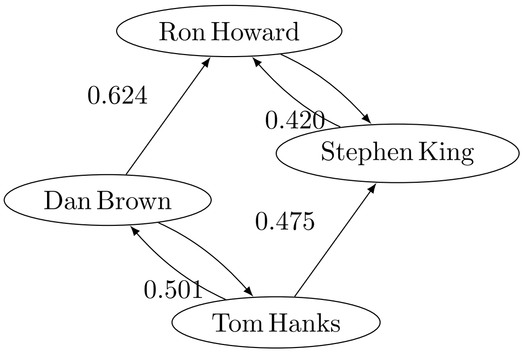


Figure 4. The corresponding articles for actor "Tom Hanks" and director "Ron Howard" are shared by fiction authors "Dan Brown" and "Stephen King".

ticles linked to each other. A few screenshots from our planned mobile application are shown in Figure 3.

5. Illustrative Example

In this section, we provide an illustrative example, elaborating on the images from Figure 3. We take the corresponding articles for authors "Dan Brown" and "Stephen King" together with their keyword-articles. The two authors have many shared links and backlinks, namely 11 shared links and 27 shared backlinks. Compared to the high number of links and backlinks that each has, 122 links + 344 backlinks for Dan Brown and 442 links + 2044 backlinks for Stephen King, their shared ones are few, but sufficient and significant. Table 1 lists some of them together with their relatedness scores, computed using Equation 7 for article-to-article relatedness and Equation 14 for article-to-pair. This situation applies to most pairs of authors, preventing the recommender system from displaying many results.

From the lists of shared links and backlinks we selected two representative keyword–articles that also happen to be among the highest in relatedness with both authors, as shown in Figure 4. Their relatedness coefficients are computed using Equation 7, which we designed to take into account the entire linking structure of compared articles. Note that these coefficients do not belong to link-edges, which can be reciprocal or even missing, but to the compared articles. In Figure 4, we just added these coefficients between the articles for which we computed relatedness.

There are of course a few articles that relate significantly more to one author than to the other. For example, "The Da Vinci Code", novel written by Dan Brown, relates 0.978 to Dan Brown, as expected, and only 0.427 to Stephen King. It is arguable whether "The Da Vinci Code" should be given more importance, say, than "Tom Hanks". Therefore, applying Equation 14 to compute how related "The Da Vinci Code" and "Tom Hanks" each are to the pair of articles ("Dan Brown", "Stephen King"), we obtain for "The Da Vinci Code" 0.306 and for "Tom Hanks" 0.462, concluding that "Tom Hanks" relates more to the two authors taken as pair than "The Da Vinci Code". In Table 1 the relatedness with the pair, computed using Equation 14, is shown in the last column.

In the example from Figure 4, the actor Tom Hanks and the director Ron Howard score quite well for the pair Dan Brown and Stephen King. This means that if a user likes Stephen King and then also likes Tom Hanks, he or she receives Dan Brown as recommendation with the reasoning that Tom Hanks played the leading role in Dan Brown's "The Da Vinci Code" movie trilogy, and also played the leading role in Stephen King's movie adaptation "The Green Mile". Similarly, Ron Howard directed the movies from "The Da Vinci Code" series, and is currently directing a TV series, "The Dark Tower", written by Stephen King. Their relatedness scores are quite similar, though it is interesting to observe, despite their similarity, that one is a shared backlink, while the other is a shared link. This illustrates that Wikipedia does not follow a hierarchical structure, links and backlinks being equally valuable as keywords.

Additionally, by looking solely at this graph, it can be observed that the PageRank of "Ron Howard" is equal to the one of 'Stephen King' and higher than the PageRank of the other two articles. To be more precise, "Ron Howard" and "Stephen King" would both have a PageRank of 1.333, whereas the PageRanks for "Dan Brown" and "Tom Hanks" would be 0.667 each. In this case, the *weighted average* from Equation 6 required to compute relatedness between "Dan Brown" and "Stephen King" corresponds to an arithmetic average, α being equal to 0.5. After also applying Equation 7, this relatedness has a coefficient of 0.503.

6. Conclusion and Future Work

We have shown throughout this paper a useful method to compute relatedness between nodes in a graph and to implement it in a recommender system. We used Wikipedia as the knowledge base and we exemplified our recommendations on thriller/crime fiction authors. We demonstrated that our approach has significant advantages over more classical approaches such as collaborative filtering. We also adapted and improved the relatedness measurements from related research, taking full advantage of the linking structure and at the same time keeping computation inexpensive. Finally, we discussed the *surprise level*, a feature of the recommender that allows the user to choose how surprising the results should be; and we also presented the *relevance factor*, allowing a better assessment of shared keyword-articles between selected and resulting authors.

As this is an ongoing research, future work involves further evaluation methods. We are currently assessing our relatedness algorithm and its impact on the performance of the recommender system and the surprise level, by comparing it with relatedness based on fewer linking properties, such as shared links and backlinks. We also plan to evaluate our recommender system against leading algorithms used on very large userbases, in order to measure how similar the results are. Furthermore, we intend to test the recommender on both expert users and unknowledgeable users, in order to assess their satisfaction with our approach. Finally, we work towards launching the website and mobile application intended for public use.

Acknowledgments

The fourth author is supported by the NWO COMPASS project (grant #612.065.92).

References

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, *19*, 370–383.

Dolan, S. (2011). Six degrees of Wikipedia. `http://www.netsoc.tcd.ie/~mu/wiki`.

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *Proceedings of the 20th international joint conference on Artifical intelligence* (pp. 1606–1611).

Jeh, G., & Widom, J. (2002). Simrank: A measure of structural-context similarity. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 538–543).

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal ACM*, *46*, 604–632.

Langville, A. N., & Meyer, C. D. (2006). *Google's PageRank and beyond: The science of search engine rankings.* Princeton University Press.

Lin, Z., Lyu, M. R., & King, I. (2007). Extending link-based algorithms for similar web pages with neighborhood structure. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 263–266).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval.* Cambridge University Press.

Melville, P., & Sindhwani, V. (2010). Recommender systems. In *Encyclopedia of machine learning*, chapter 705, 829–838. Boston, MA.

Milne, D., & Witten, I. H. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *Proceedings of the First AAAI Workshop on Wikipedia and Artificial Intelligence* (pp. 25–30).

Onuma, K., Tong, H., & Faloutsos, C. (2009). Tangent: A novel, 'Surprise me', recommendation algorithm. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 657–666).

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Eetrieval* (pp. 253–260).

Wikipedia (2011a). Help: What links here. `http://en.wikipedia.org/wiki/Help:What_links_here`.

Wikipedia (2011b). Wikipedia: About. `http://en.wikipedia.org/wiki/Wikipedia:About`.

Wikipedia (2011c). Wikipedia: Manual of style (linking). `http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)`.