Improving Diversity in Ranking using Absorbing Random Walks

Xiaojin Zhu **Andrew B. Goldberg** **Jurgen Van Gael** **David Andrzejewski**
Department of Computer Sciences
University of Wisconsin, Madison
Madison, WI 53705
{jerryzhu, goldberg, jvangael, andrzeje}@cs.wisc.edu

Abstract

We introduce a novel ranking algorithm called GRASSHOPPER, which ranks items with an emphasis on diversity. That is, the top items should be different from each other in order to have a broad coverage of the whole item set. Many natural language processing tasks can benefit from such diversity ranking. Our algorithm is based on random walks in an absorbing Markov chain. We turn ranked items into absorbing states, which effectively prevents redundant items from receiving a high rank. We demonstrate GRASSHOPPER's effectiveness on extractive text summarization: our algorithm ranks between the 1st and 2nd systems on DUC 2004 Task 2; and on a social network analysis task that identifies movie stars of the world.

1 Introduction

Many natural language processing tasks involve ranking a set of items. Sometimes we want the top items to be not only good individually but also *diverse* collectively. For example, extractive text summarization generates a summary by selecting a few good sentences from one or more articles on the same topic (Goldstein et al., 2000). This can be formulated as ranking all the sentences, and taking the top ones. A good sentence is one that is representative, i.e., similar to many other sentences, so that it likely conveys the central meaning of the articles. On the other hand, we do not want multiple near-identical sentences. The top sentences should be diverse.

As another example, in information retrieval on news events, an article is often published by multiple newspapers with only minor changes. It is undesirable to rank all copies of the same article highly, even though it may be the most relevant. Instead, the top results should be different and complementary. In other words, one wants 'subtopic diversity' in retrieval results (Zhai et al., 2003).

The need for diversity in ranking is not unique to natural language processing. In social network analysis, people are connected by their interactions, e.g., phone calls. Active groups of people have strong interactions among them, but many groups may exist with fewer interactions. If we want a list of people that represent various groups, it is important to consider both activity and diversity, and not to fill the list with people from the same active groups.

Given the importance of diversity in ranking, there has been significant research in this area. Perhaps the most well-known method is maximum marginal relevance (MMR) (Carbonell and Goldstein, 1998), as well as cross-sentence informational subsumption (Radev, 2000), mixture models (Zhang et al., 2002), subtopic diversity (Zhai et al., 2003), diversity penalty (Zhang et al., 2005), and others. The basic idea is to penalize redundancy by lowering an item's rank if it is similar to items already ranked. However, these methods often treat centrality ranking and diversity ranking separately, sometimes with heuristic procedures.

Proceedings of NAACL HLT 2007, pages 97–104,
Rochester, NY, April 2007.

We propose GRASSHOPPER (**G**raph **R**andom-walk with **A**bsorbing **S**tate**S** that **HOP**s among **PE**aks for **R**anking), a novel ranking algorithm that encourages diversity. GRASSHOPPER is an alternative to MMR and variants, with a principled mathematical model and strong empirical performance. It ranks a set of items such that: 1. A highly ranked item is representative of a local group in the set, i.e., it is similar to many other items (**centrality**); 2. The top items cover as many distinct groups as possible (**diversity**); 3. It incorporates an arbitrary pre-specified ranking as prior knowledge (**prior**). Importantly GRASSHOPPER achieves these in a unified framework of *absorbing Markov chain random walks*. The key idea is the following: We define a random walk on a graph over the items. Items which have been ranked so far become absorbing states. These absorbing states 'drag down' the importance of similar unranked states, thus encouraging diversity. Our model naturally balances centrality, diversity, and prior. We discuss the algorithm in Section 2. We present GRASSHOPPER's empirical results on text summarization and social network analysis in Section 3.

2 The GRASSHOPPER Algorithm

2.1 The Input

GRASSHOPPER requires three inputs: a graph W, a probability distribution $\mathbf{r}$ that encodes the prior ranking, and a weight $\lambda \in [0, 1]$ that balances the two.

The user needs to supply a graph with n nodes, one for each item. The graph is represented by an $n \times n$ weight matrix W, where w_{ij} is the weight on the edge from i to j. It can be either directed or undirected. W is symmetric for undirected graphs. The weights are non-negative. The graph does not need to be fully connected: if there is no edge from item i to j, then $w_{ij} = 0$. Self-edges are allowed. For example, in text summarization one can create an undirected, fully connected graph on the sentences. The edge between sentences i, j has weight w_{ij}, their cosine similarity. In social network analysis one can create a directed graph with w_{ij} being the number of phone calls i made to j. The graph should be constructed carefully to reflect domain knowledge. For examples, see (Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Pang and Lee, 2004).

The user can optionally supply an arbitrary ranking on the items as prior knowledge. In this case GRASSHOPPER can be viewed as a re-ranking method. For example, in information retrieval, the prior ranking can be the ranking by relevance scores. In text summarization, it can be the position of sentences in the original article. (There is evidence that the first few sentences in an article are likely good summaries.) Somewhat unconventionally, the prior ranking is represented as a probability distribution $\mathbf{r} = (r_1, \cdots, r_n)^\top$ such that $r_i \geq 0, \sum_{i=1}^n r_i = 1$. The highest-ranked item has the largest probability, the next item has smaller probability, and so on. A distribution gives the user more control. For example $\mathbf{r}_a = (0.1, 0.7, 0.2)^\top$ and $\mathbf{r}_b = (0.3, 0.37, 0.33)^\top$ both represent the same ranking of items 2, 3, 1, but with different strengths. When there is no prior ranking, one can let $\mathbf{r} = (1/n, \cdots, 1/n)^\top$, the uniform distribution.

2.2 Finding the First Item

We find the first item in GRASSHOPPER ranking by teleporting random walks. Imagine a random walker on the graph. At each step, the walker may do one of two things: with probability λ, she moves to a neighbor state[1] according to the edge weights; otherwise she is teleported to a random state according to the distribution $\mathbf{r}$. Under mild conditions (which are satisfied in our setting, see below), the stationary distribution of the random walk defines the visiting probabilities of the nodes. The states with large probabilities can be regarded as central items, an idea used in Google PageRank (Page et al., 1998) and other information retrieval systems (Kurland and Lee, 2005; Zhang et al., 2005), text summarization (Erkan and Radev, 2004), keyword extraction (Mihalcea and Tarau, 2004) and so on. Depending on λ, items high on the user-supplied prior ranking $\mathbf{r}$ may also have large stationary probabilities, which is a way to incorporate the prior ranking.

As an example, we created a toy data set with 300 points in Figure 1(a). There are roughly three groups with different densities. We created a fully connected graph on the data, with larger edge weights if points are closer[2]. Figure 1(b) shows the stationary distribution of the random walk on the graph.

[1] We use *state*, *node* and *item* interchangeably.

[2] We use $w_{ij} = \exp(-\|x_i - x_j\|^2/0.16)$, $\lambda = 1$.

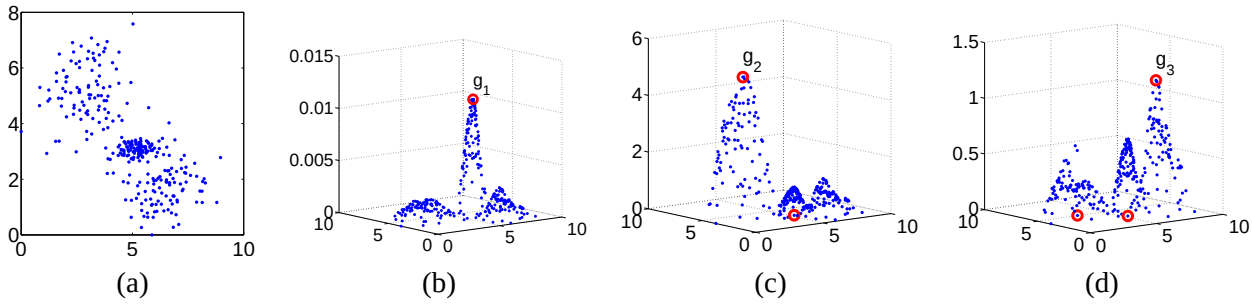


Figure 1: (a) A toy data set. (b) The stationary distribution π reflects centrality. The item with the largest probability is selected as the first item g_1. (c) The expected number of visits $\mathbf{v}$ to each node after g_1 becomes an absorbing state. (d) After both g_1 and g_2 become absorbing states. Note the diversity in g_1, g_2, g_3 as they come from different groups.

Items at group centers have higher probabilities, and tighter groups have overall higher probabilities.

However, the stationary distribution does not address diversity at all. If we were to rank the items by their stationary distribution, the top list would be dominated by items from the center group in Figure 1(b). Therefore we only use the stationary distribution to find the first item, and use a method described in the next section to rank the remaining items.

Formally we first define an $n \times n$ raw transition matrix $\tilde{P}$ by normalizing the rows of W: $\tilde{P}_{ij} = w_{ij} / \sum_{k=1}^{n} w_{ik}$, so that $\tilde{P}_{ij}$ is the probability that the walker moves to j from i. We then make the walk a teleporting random walk P by interpolating each row with the user-supplied initial distribution $\mathbf{r}$:

$$P = \lambda \tilde{P} + (1 - \lambda)\mathbf{1}\mathbf{r}^{\top}, \qquad (1)$$

where $\mathbf{1}$ is an all-1 vector, and $\mathbf{1}\mathbf{r}^{\top}$ is the outer product. If $\lambda < 1$ and $\mathbf{r}$ does not have zero elements, our teleporting random walk P is irreducible (possible to go to any state from any state by teleporting), aperiodic (the walk can return to a state after any number of steps), all states are positive recurrent (the expected return time to any state is finite) and thus ergodic (Grimmett and Stirzaker, 2001). Therefore P has a unique stationary distribution $\pi = P^{\top}\pi$. We take the state with the largest stationary probability to be the first item g_1 in GRASSHOPPER ranking: $g_1 = \operatorname{argmax}_{i=1}^{n} \pi_i$.

2.3 Ranking the Remaining Items

As mentioned early, the key idea of GRASSHOPPER is to turn ranked items into absorbing states. We first turn g_1 into an absorbing state. Once the random walk reaches an absorbing state, the walk is absorbed and stays there. It is no longer informative to compute the stationary distribution of an absorbing Markov chain, because the walk will eventually be absorbed. Nonetheless, it is useful to compute the *expected number of visits* to each node before absorption. Intuitively, those nodes strongly connected to g_1 will have many fewer visits by the random walk, because the walk tends to be absorbed soon after visiting them. In contrast, groups of nodes far away from g_1 still allow the random walk to linger among them, and thus have more visits. In Figure 1(c), once g_1 becomes an absorbing node (represented by a circle 'on the floor'), the center group is no longer the most prominent: nodes in this group have fewer visits than the left group. Note now the y-axis is the number of visits instead of probability.

GRASSHOPPER selects the second item g_2 with the largest expected number of visits in this absorbing Markov chain. This naturally inhibits items similar to g_1 and encourages diversity. In Figure 1(c), the item near the center of the left group is selected as g_2. Once g_2 is selected, it is converted into an absorbing state, too. This is shown in Figure 1(d). The right group now becomes the most prominent, since both the left and center groups contain an absorbing state. The next item g_3 in ranking will come from the right group. Also note the range of y-axis is smaller:

with more absorbing states, the random walk will be absorbed sooner. The procedure is repeated until all items are ranked. The name GRASSHOPPER reflects the 'hopping' behavior on the peaks.

It is therefore important to compute the expected number of visits in an absorbing Markov chain. Let G be the set of items ranked so far. We turn the states $g \in G$ into absorbing states by setting $P_{gg} = 1$ and $P_{gi} = 0, \forall i \neq g$. If we arrange items so that ranked ones are listed before unranked ones, we can write P as

$$P = \begin{bmatrix} \mathbf{I}_G & \mathbf{0} \\ R & Q \end{bmatrix}. \quad (2)$$

Here $\mathbf{I}_G$ is the identity matrix on G. Submatrices R and Q correspond to rows of unranked items, those from (1). It is known that the *fundamental matrix*

$$N = (\mathbf{I} - Q)^{-1} \quad (3)$$

gives the expected number of visits in the absorbing random walk (Doyle and Snell, 1984). In particular N_{ij} is the expected number of visits to state j before absorption, if the random walk started at state i. We then average over all starting states to obtain v_j, the expected number of visits to state j. In matrix notation,

$$\mathbf{v} = \frac{N^\top \mathbf{1}}{n - |G|}, \quad (4)$$

where $|G|$ is the size of G. We select the state with the largest expected number of visits as the next item $g_{|G|+1}$ in GRASSHOPPER ranking:

$$g_{|G|+1} = \text{argmax}_{i=|G|+1}^{n} \, v_i. \quad (5)$$

The complete GRASSHOPPER algorithm is summarized in Figure 2.

2.4 Some Discussions

To see how λ controls the tradeoff, note when $\lambda = 1$ we ignore the user-supplied prior ranking $\mathbf{r}$, while when $\lambda = 0$ one can show that GRASSHOPPER returns the ranking specified by $\mathbf{r}$.

Our data in Figure 1(a) has a cluster structure. Many methods have exploited such structure, e.g., (Hearst and Pedersen, 1996; Leuski, 2001; Liu and Croft, 2004). In fact, a heuristic algorithm is to first cluster the items, then pick the central items from each cluster in turn. But it can be difficult to determine the appropriate number and control the shape of clusters. In contrast, GRASSHOPPER does *not* involve clustering. However it is still able to automatically take advantage of cluster structures in the data.

Input: W, $\mathbf{r}$, λ

1. Create the initial Markov chain P from $W, \mathbf{r}, \lambda$ (1).
2. Compute P's stationary distribution π. Pick the first item $g_1 = \text{argmax}_i \, \pi_i$.
3. Repeat until all items are ranked:
   (a) Turn ranked items into absorbing states (2).
   (b) Compute the expected number of visits $\mathbf{v}$ for all remaining items (4). Pick the next item $g_{|G|+1} = \text{argmax}_i \, v_i$

Figure 2: The GRASSHOPPER algorithm

In each iteration we need to compute the fundamental matrix (3). This involves inverting an $(n - |G|) \times (n - |G|)$ matrix, which is expensive. However the Q matrix is reduced by one row and one column in every iteration, but is otherwise unchanged. This allows us to apply the matrix inversion lemma (Sherman-Morrison-Woodbury formula) (Press et al., 1992). Then we only need to invert the matrix once in the first iteration, but not in subsequent iterations. Space precludes a full discussion, but we point out that it presents a significant speed up. A Matlab implementation can be found at `http://www.cs.wisc.edu/~jerryzhu/pub/grasshopper.m`.

3 Experiments

3.1 Text Summarization

Multi-document extractive text summarization is a prime application for GRASSHOPPER. In this task, we must select and rank sentences originating from a set of documents about a particular topic or event. The goal is to produce a summary that includes all the relevant facts, yet avoids repetition that may result from using similar sentences from multiple documents. In this section, we demonstrate that

GRASSHOPPER's balance of centrality and diversity makes it successful at this task. We present empirical evidence that GRASSHOPPER achieves results competitive with the top text summarizers in the 2004 Document Understanding Conference (http://duc.nist.gov). DUC is a yearly text summarization community evaluation, with several tasks in recent years concentrating on multi-document summarization (described in more detail below).

Many successful text summarization systems achieve a balance between sentence centrality and diversity in a two-step process. Here we review the LexRank system (Erkan and Radev, 2004), which is most similar to our current approach. LexRank works by placing sentences in a graph, with edges based on the lexical similarity between the sentences (as determined by a cosine measure). Each sentence is then assigned a centrality score by finding its probability under the stationary distribution of a random walk on this graph. Unlike the similar PageRank algorithm (Page et al., 1998), LexRank uses an undirected graph of sentences rather than Web pages, and the edge weights are either cosine values or 0/1 with thresholding. The LexRank centrality can be combined with other centrality measures, as well as sentence position information. After this first step of computing centrality, a second step performs re-ranking to avoid redundancy in the highly ranked sentences. LexRank uses cross-sentence informational subsumption (Radev, 2000) to this end, but MMR (Carbonell and Goldstein, 1998) has also been widely used in the text summarization community. These methods essentially disqualify sentences that are too lexically similar to sentences ranked higher by centrality. In short, similar graph-based approaches to text summarization rely on two distinct processes to measure each sentence's importance and ensure some degree of diversity. GRASSHOPPER, on the other hand, achieves the same goal in a unified procedure.

We apply GRASSHOPPER to text summarization in the following manner. Our graph contains nodes for all the sentences in a document set. We used the Clair Library (http://tangra.si.umich.edu/clair/clairlib) to split documents into sentences, apply stemming, and create a cosine matrix for the stemmed sentences. Cosine values are computed using TF-IDF vectors. As in LexRank, edges in the graph correspond to text similarity. To create a sparse graph, we use the cosine threshold value of 0.1 obtained in (Erkan and Radev, 2004). Specifically, the edge weight between sentence vectors s_i and s_j is defined as

$$w_{ij} = \begin{cases} 1 & \text{if } \frac{s_i^\top s_j}{\|s_i\| \cdot \|s_j\|} > 0.1 \\ 0 & \text{otherwise} \end{cases} . \quad (6)$$

The second input for GRASSHOPPER is an initial ranking distribution, which we derive from the position of each sentence in its originating document. Position forms the basis for lead-based summaries (i.e., using the first N sentences as the summary) and leads to very competitive summaries (Brandow et al., 1995). We form an initial ranking for each sentence by computing $p^{-\alpha}$, where p is the position of the sentence in its document, and α is a positive parameter trained on a development dataset. We then normalize over all sentences in all documents to form a valid distribution $r \propto p^{-\alpha}$ that gives high probability to sentences closer to the beginning of documents. With a larger α, the probability assigned to later sentences decays more rapidly.

To evaluate GRASSHOPPER, we experimented with DUC datasets. We train our parameters (α and λ) using the DUC 2003 Task 2 data. This dataset contains 30 document sets, each with an average of 10 documents about a news event. We test GRASSHOPPER's performance on the DUC 2004 Task 2, Tasks 4a and 4b data. DUC 2004 Task 2 has 50 document sets of 10 documents each. Tasks 4a and 4b explored cross-lingual summarization. These datasets consist of Arabic-to-English translations of news stories. The documents in Task 4a are machine-translated, while Task 4b's are manually-translated. Note that we handle the translated documents in exactly the same manner as the English documents.

We evaluate our results using the standard text summarization metric ROUGE (http://www.isi.edu/~cyl/ROUGE/). This is a recall-based measure of text co-occurrence between a machine-generated summary and model summaries manually created by judges. ROUGE metrics exist based on bigram, trigram, and 4-gram overlap, but ROUGE-1 (based on unigram matching) has been found to correlate best with human judgments (Lin and Hovy, 2003).

Using the DUC 2003 training data, we tuned α and λ on a small grid ($\alpha \in \{0.125, 0.25, 0.5, 1.0\}$; $\lambda \in \{0.0, 0.0625, 0.125, 0.25, 0.5, 0.95\}$). Specifically, for each of the 30 DUC 2003 Task 2 document sets, we computed ROUGE-1 scores comparing our generated summary to 4 model summaries. We averaged the resulting ROUGE-1 scores across all 30 sets to produce a single average ROUGE-1 score to assess a particular parameter configuration. After examining the results for all 24 configurations, we selected the best one: $\alpha = 0.25$ and $\lambda = 0.5$.

Table 1 presents our results using these parameter values to generate summaries for the three DUC 2004 datasets. Note that the averages listed are actually averages over 4 model summaries per set, and over all the sets. Following the standard DUC protocol, we list the confidence intervals calculated by ROUGE using a bootstrapping technique. The final column compares our results to the official systems that participated in the DUC 2004 evaluation. GRASSHOPPER is highly competitive in these text summarization tasks: in particular it ranks between the 1st and 2nd automatic systems on 2004 Task 2. The lower performance in Task 4a is potentially due to the documents being machine-translated. If they contain poorly translated sentences, graph edges based on cosine similarity could be less meaningful. For such a task, more advanced text processing is probably required.

3.2 Social Network Analysis

As another application of GRASSHOPPER, we identify the nodes in a social network that are the most prominent, and at the same time maximally cover the network. A node's prominence comes from its intrinsic stature, as well as the prominence of the nodes it touches. However, to ensure that the top-ranked nodes are representative of the larger graph structure, it is important to make sure the results are not dominated by a small group of highly prominent nodes who are closely linked to one another. This requirement makes GRASSHOPPER a useful algorithm for this task.

We created a dataset from the Internet Movie Database (IMDb) that consists of all comedy movies produced between 2000 and 2006, and have received more than 500 votes by IMDb users. This results in 1027 movies. We form a social network of actors by co-star relationship. Not surprisingly, actors from the United States dominate our dataset, although a total of 30 distinct countries are represented. We seek an actor ranking such that the top actors are prominent. However, we also want the top actors to be diverse, so they represent comedians from around the world.

This problem is framed as a GRASSHOPPER ranking problem. For each movie, we considered only the main stars, i.e., the first five cast members, who tend to be the most important. The resulting list contains 3452 unique actors. We formed a social network where the nodes are the actors, and undirected weighted edges connect actors who have appeared in a movie together. The edge weights are equal to the number of movies from our dataset in which both actors were main stars. Actors are also given a self-edge with weight 1. The co-star graph is given to GRASSHOPPER as an input. For the prior actor ranking, we simply let r be proportional to the number of movies in our dataset in which an actor has appeared. We set the weight $\lambda = 0.95$. It is important to note that no country information is ever given to GRASSHOPPER.

We use two measurements, 'country coverage' and 'movie coverage', to study the diversity and prominence of the ranking produced by GRASSHOPPER. We compare GRASSHOPPER to two baselines: ranking based solely on the number of movies an actor has appeared in, MOVIECOUNT, and a randomly generated ranking, RANDOM.

First, we calculate 'country coverage' as the number of different countries represented by the top k actors, for all k values. Each actor represents a single country—the country that the actor has appeared in the most. We hypothesize that actors are more likely to have co-star connections to actors within the same country, so our social network may have, to some extent, a clustering structure by country. 'Country coverage' approximates the number of clusters represented at different ranks.

Figure 3(a) shows that country coverage grows much more rapidly for GRASSHOPPER than for MOVIECOUNT. That is, we see more comedians from around the world ranked highly by GRASSHOPPER. In contrast, the top ranks of MOVIECOUNT are dominated by US actors, due to the relative abundance of US movies on IMDb. Many other countries are

Dataset	Number of Doc. Sets	Average ROUGE-1	95% C.I.	GRASSHOPPER Unofficial Rank
DUC 2004 Task 2	50	0.3755	[0.3622, 0.3888]	Between 1 & 2 of 34
DUC 2004 Task 4a	24	0.3785	[0.3613, 0.3958]	Between 5 & 6 of 11
DUC 2004 Task 4b	24	0.4067	[0.3883, 0.4251]	Between 2 & 3 of 11

Table 1: Text summarization results on DUC 2004 datasets. GRASSHOPPER was configured using parameters tuned on the DUC 2003 Task 2 dataset. The rightmost column lists what our rank would have been if we had participated in the DUC 2004 evaluation.

not represented until further down in the ranked list. This demonstrates that GRASSHOPPER ranking is successful in returning a more diverse ranking. Because of the absorbing states in GRASSHOPPER, the first few highly ranked US actors encourage the selection of actors from other regions of the co-star graph, which roughly correspond to different countries. RANDOM achieves even higher country coverage initially, but is quickly surpassed by GRASSHOPPER. The initial high coverage comes from the random selection of actors. However these randomly selected actors are often not prominent, as we show next.

Second, we calculate 'movie coverage' as the total number of unique movies the top k actors are in. We expect that actors who have been in more movies are more prominent. This is reasonable because we count an actor in a movie only if the actor is among the top five actors from that movie. Our counts thus exclude actors who had only small roles in numerous movies. Therefore high movie coverage roughly corresponds to ranking more prominent actors highly. It is worth noting that this measure also partially accounts for diversity, since an actor whose movies completely overlap with those of higher-ranked actors contributes nothing to movie coverage (i.e., his/her movies are already covered by higher-ranked actors).

Figure 3(b) shows that the movie coverage of GRASSHOPPER grows more rapidly than MOVIECOUNT, and much more rapidly than RANDOM. The results show that, while the RANDOM ranking is diverse, it is not of high quality because it fails to include many prominent actors in its high ranks. This is to be expected of a random ranking. Since the vast majority of the actors appear in only one movie, the movie coverage curve is roughly linear in the number of actors. By ranking more prominent actors highly, the GRASSHOPPER and MOVIECOUNT movie coverage curves grow faster. Many of the US actors highly ranked by MOVIECOUNT are co-stars of one another, so GRASSHOPPER outperforms MOVIECOUNT in terms of movie coverage too.

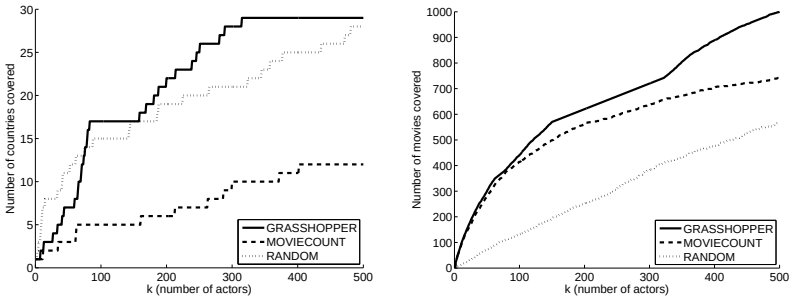
We inspect the GRASSHOPPER ranking, and find the top 5 actors to be Ben Stiller, Anthony Anderson, Johnny Knoxville, Eddie Murphy and Adam Sandler. GRASSHOPPER also brings many countries, and major stars from those countries, into the high ranks. Examples include Mads Mikkelsen ("synonym to the great success the Danish film industry has had"), Cem Yilmaz ("famous Turkish comedy actor, caricaturist and scenarist"), Jun Ji-Hyun ("face of South Korean cinema"), Tadanobu Asano ("Japan's answer to Johnny Depp"), Aamir Khan ("prominent Bollywood film actor"), and so on[3]. These actors are ranked significantly lower by MOVIECOUNT.

These results indicate that GRASSHOPPER achieves both prominence and diversity in ranking actors in the IMDb co-star graph.

4 Conclusions

GRASSHOPPER ranking provides a unified approach for achieving both diversity and centrality. We have shown its effectiveness in text summarization and social network analysis. As future work, one direction is "partial absorption," where at each absorbing state the random walk has an escape probability to continue the random walk instead of being absorbed. Tuning the escape probability creates a continuum between PageRank (if the walk always escapes) and GRASSHOPPER (if always absorbed). In addition, we will explore the issue of parameter learning, and

[3]Quotes from IMDb and Wikipedia.



(a) Country coverage (b) Movie coverage

Figure 3: (a) Country coverage at ranks up to 500, showing that GRASSHOPPER and RANDOM rankings are more diverse than MOVIECOUNT. (b) Movie coverage at ranks up to 500, showing that GRASSHOPPER and MOVIECOUNT have more prominent actors than RANDOM. Overall, GRASSHOPPER is the best.

user feedback (e.g., "This item should be ranked higher."). We also plan to apply GRASSHOPPER to a variety of tasks, including information retrieval (for example ranking news articles on the same event as in Google News, where many newspapers might use the same report and thus result in a lack of diversity), image collection summarization, and social network analysis for national security and business intelligence.

Acknowledgment We thank Mark Craven and the anonymous reviewers for helpful comments. This work is supported in part by Wisconsin Alumni Research Foundation (WARF) and NLM training grant 5T15LM07359.

References

R. Brandow, K. Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Inf. Process. Manage.*, 31(5):675–685.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98*.

P.G. Doyle and J.L. Snell. 1984. *Random Walks and Electric Networks*. Mathematical Assoc. of America.

Güneş Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48.

Geoffrey R. Grimmett and David R. Stirzaker. 2001. *Probability and Random Processes*. Oxford Science Publications, third edition.

Marti A. Hearst and Jan O. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *SIGIR-96*.

Oren Kurland and Lillian Lee. 2005. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *SIGIR'05*.

Anton Leuski. 2001. Evaluating document clustering for interactive information retrieval. In *CIKM'01*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL'03*, pages 71–78.

Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *SIGIR'04*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *EMNLP'04*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, pages 271–278.

W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 1992. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press New York, NY, USA.

Dragomir Radev. 2000. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*.

ChengXiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *SIGIR'03*.

Yi Zhang, Jamie Callan, and Thomas Minka. 2002. Novelty and redundancy detection in adaptive filtering. In *SIGIR'02*.

Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. 2005. Improving web search results using affinity graph. In *SIGIR'05*.