

# Bee Hive At Work: Story Tracking Case Study

Pavol Navrat, Lucia Jastrzemska, Tomas Jelinek  
Faculty of Informatics and Information Technologies  
Slovak University of Technology  
Bratislava, Slovakia

Email: navrat@fiit.stuba.sk, jastrzemska@gmail.com, tj.jelinek@gmail.com

**Abstract**—Information can change rapidly on the web. For example, news may hint some new story starts to develop. Many more news related to the original event begin to pour in the web. Imagine a person interested in how the story develops. It may be very difficult to trace it by trying to find the most relevant pages with most recent news on it. Our goal is to support user who wants to keep track of a developing story. We propose an approach and a system based on a bee hive model. The problem we focus on in this paper is that it is not possible to download all the pages using e.g. the breadth-first algorithm, nor to constantly revisit all the pages to see if new information were added. We propose to use a focused crawler to download the pages. With a prototype of our system, we performed a case study that shows that the system is able to collect relevant pages, it can monitor the story being developed during the search and it can even reconstruct the story backwards in time.

**Index Terms**—story tracking; bee hive model; web crawler; web search;

## I. INTRODUCTION

Information on the Internet is growing fast. This can be particularly visible in electronic newspapers, where new information is being constantly added and changed.

Finding and reading most relevant and up to date articles about the specific topic has been made potentially much more easier and difficult at the same time with the advent of the web. On the web, it requires continuously observing all the news sources for updates of stories one is interested in. All this, including also discovering new data sources can be problematic if not impossible for a human. Therefore a system capable of doing these tasks might be helpful. Input of such a system would be keywords describing somehow the story of interest and the system would run until the story is developing. As an output the user would get the story related to this issue as recorded in a series of data sources (articles, documents on web pages). We design this system to be running on the user computer. The problem we focus on in this paper is that it is not possible to download all the pages using e.g. the breadth-first algorithm, nor to constantly revisit all the pages to see if new information were added. We propose to use a focused crawler to download the pages. We took the inspiration for constructing the crawler from nature, particularly the social behaviour of honey bees.

## II. RELATED WORK

The field of focused crawlers is not new. Since the early years of Internet there was effort to reduce the amount of pages crawled to the most relevant ones. The early concepts

of such crawlers include best-first, fish search [4] and shark search [8] algorithms. In late 90's the term focused crawler was introduced in [9].

Even the use of focused crawler for online search is not new. In system called Fetuccino [2] the authors tried to solve a classic problem of web search with offline database - that the results returned have changed since indexed into the database. They called the classical web search as *static search* and enhanced it by *dynamic search*. The dynamic search was an approach to revisit the pages at the time of searching after the results from static search had been obtained. The results were then modified according to the dynamic search and provided to the user.

There have been attempts to propose nature inspired algorithm for focused crawling. Focused Ant Crawling Algorithm [6], for hypertext graph crawling, is claimed to be better than the Shark-Search crawling algorithm.

Another example of using online search is agent InfoSpider [15]. The authors based this agent on previous works on adaptive agents [16], [13], [14] and on the assumption that the Web dynamic (noise, heterogeneity, decentralization) is similar to the natural environment where organism have to adapt and survive. These organisms demonstrates capabilities such as local adaptation and distributed management. Artificial agents, such as natural agents, should be autonomous, intelligent, distributed, they should create the „ecology” of agents.

InfoSpider agent moves from one on-line document to another by choosing the link to follow. It adapts to its environment by a neural network (used to select link to continue) and evolution of agents (agents are being created and destroyed by evolution). The aim of the adaptation is to focus on the relevant regions.

Another area related to our work is story tracking. In [17] there is published an approach of handling information overflow by clustering similar articles into stories.

## III. BEE HIVE MODEL

In the previous chapter, we described several web agents inspired by nature. We too took the inspiration in nature, namely in the foraging behaviour of honey bees (*Apis mellifera*). Honey bees have unique ability to make collective decision without the need of any central decision unit. Every single bee knows only partial information about the sources she has visited, but as a whole the hive can determine the best source of all. Moreover, the bee hive flexibly reacts to changes of

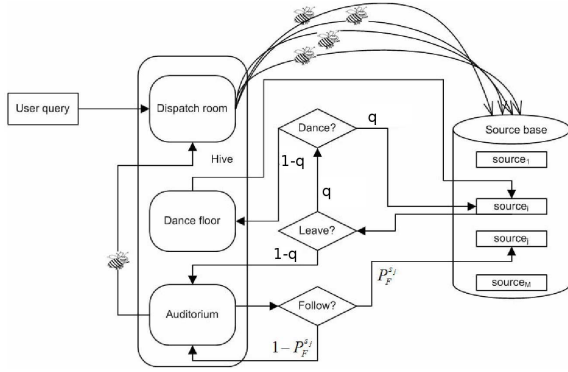


Fig. 1. The model of bee hive ([10])

its environment. Honey bees achieve this by communication. The bees behaviour was modelled by various researchers (e.g. [5], [7]) either from field of biology or computer science. We chose the [10] model (figure 1) for its simplicity and clear state transitions based on source quality.

The search for food sources starts in the dispatch room. All the bees are in the dispatch room and choose randomly from sources in it. The sources can be statically enumerated or can be generated randomly (depending on the problem domain, where the model is applied). When the bee chooses the source, she flies to visit it and to estimate its quality.

When the bee returns to the hive she either abandons the source or stays with it. When the bee decides to stay with the source she can decide whether to propagate it on the dance floor or revisit it directly. When the bee decides to propagate the source she will dance for the source for the time estimated as the quality of the source multiplied by the maximal dancing time. After this time she will return to her source. With probability equal to the number of bees dancing for the same source divided by the number of all dancing bees ( $P_f^{S_j}$ ) the bee will fly to visit the chosen source, with opposite probability she will stay in the auditorium. If the bee did not choose any source to follow within the specified amount of time, she would fly to the dispatch room and choose source from there.

This model has the following set of parameters

- *Number of bees*
- *Maximal dancing time* Maximal time the bee can stay in the dance floor (we used iterations)
- *Maximal time in auditorium* Maximal time the bee can stay in the auditorium (we used iterations)

We took this model and specified the behaviour of the bee outside the hive [11] (source code is available at [18]). The web page was used as the „source” and the aim of the hive was to find the most relevant pages and thus focus the search for new pages into the more promising areas. When the bee flies outside the hive to the source (web page), it will estimate its quality (relevancy) and with the probability  $q$  it will stay on the current page, with probability  $1 - q$  she will follow one of the links on the page to visit the new source. Then she

will with probability  $q$  fly back to the hive with her current source or with the opposite probability stay outside the hive and search for better sources. The bee cannot stay outside the hive forever, therefore we used the concept of „energy” taken from [15]. Every time the bee visits some source, the energy will increment by the quality of the source (non relevant source has zero quality) and decrement by the specified parameter. If the bee has no more energy ( $energy \leq 0$ ) she will return to the hive regardless of the other conditions.

### A. Quality calculation

We took only very simple concept of estimating the page quality (relevancy) -

The quality is divided into two components. Each of them can reach their maximal value given by the parameter. The sum of these parameters has to be equal 1.

The count quality counts the query occurrences  $n$  on the page and using the  $Q_{COUNT}$  parameter calculates the quality according to following formula:

$$q_{count} = \frac{-1}{2(n + \frac{1}{2Q_{COUNT}})} + Q_{COUNT}$$

The header quality searches all page headers and page title for query occurrence It then chooses the minimal header number (for case of  $\langle title \rangle$  tag it was 0,  $\langle h1 \rangle$  number is 1 ...  $\langle h6 \rangle$  number is 6) and calculates the quality according to the formula:

$$q_{header} = Q_{HEADER} - h * \frac{Q_{HEADER}}{HEADER_{MAX} + 1}$$

where  $Q_{HEADER}$  is the maximal value allowed for header quality,  $h$  is the minimal header number and  $HEADER_{MAX}$  is the maximal header number we want to take into account (for example if we want to ignore headers  $\langle h5 \rangle$  and  $\langle h6 \rangle$ , the number would be 5).

### B. Link Selection

The number of links on the web page can be very high - in our case study the average number of links per page was 89. We therefore employed a weighted method for choosing the link to follow. Every link was initialized to the weight 1. They were given one extra point for each of the following properties:

- the page and the link on it have the same subdomain
- the page and the link on it have the same domain
- the searched query appears in the text around the link
- the searched query appears in link description

In the end all the weights are normalized to interval  $\langle 0,1 \rangle$  and the link to follow is chosen by the roulette wheel.

#### IV. BEE SCOUTS

While performing the experiments with this behaviour we encountered a problem with discovering of few relevant sources where the bees could start the search. We again found inspiration in nature [3] and in the failed follower hypothesis [1]. The foraging bees fall within one of two categories - *scouts* or *recruits*. Scouts search for food independently regardless of the other bees. Recruits are bees that have followed the dance of some other bee. Under the failed follower hypothesis the scouts are „failed followers”. It means that if the bee does not find the dancing bee to follow, it will become a scout and search for food on her own. As a result, if the food is scarce, the probability of finding a dancing bee is low and more bees become scouts. If there is plenty of food, there will be more dancing bees and consequently more bees become recruits. We integrated this hypothesis and our original model without even modifying it. We can consider the bees in auditorium that have chosen some source from dancing bees as recruits. Their behaviour outside the hive is the same as proposed. The bees that have failed to choose the source in auditorium (e. g because the food is scarce and no bees are dancing) goes according to the model to the dispatch room. We can consider every bee flying out of the dispatch room as scout. The behaviour of scouts outside the hive is similar to regular recruit with one exception, if the bee finds the source with non zero quality, she will directly go to the dance room to propagate it (after dancing she will become recruit).

#### V. STORY TRACKING

We used the model described above to perform story tracking. Aim of the on-line search is not the single information, the aim is to find a relevant set of pages which would create a story. This should not be a general search engine. Instead, it is supposed to be used on news portals or any other sites containing frequently changing or added information.

The best application is for headline stories where the new information is being added very often. This search would run several hours or days (while the story is developing) and the user would watch the story to evolve.

An example of such a headline story are elections, which are usually closely observed by the media and public. In the next chapter we present a case study of story tracking the recent presidential elections in Slovakia.

#### VI. CASE STUDY

The aim of the case study was to explore if the algorithm can track a story being in development. We chose the second round of presidential elections in Slovakia, with two candidates Mrs. Iveta Radičová and Mr. Ivan Gašparovič. The story tracking started on the day of the elections on 4<sup>th</sup> April 2009 at 3 p.m and was completed next morning at 7. am. The algorithm should have tracked what the candidates did on the elections day, the first results announcement and the first reactions.

#### A. Settings

The search started from two Slovak news portals [www.sme.sk](http://www.sme.sk) and [www.pravda.sk](http://www.pravda.sk). We searched for news contributing to a developing story either from keywords (in Slovak language) *results of elections*, *Radičová* and *Gašparovič*. We used parameters and settings from the table I.

TABLE I  
PARAMETERS USED IN CASE STUDY.

<b>number of bees</b>	20
<b>maximal dancing time</b>	10 iterations
<b>maximal time in auditorium</b>	5 iterations
<b>default energy</b>	1
<b>energy decrement</b>	0.05
<b>max count quality</b>	0.8
<b>max header quality</b>	0.2
<b>max header number</b>	3

#### B. Statistical Results

During the experiment the algorithm discovered 4615 different pages from various domains, 742 of them had non zero quality. When we explored these results manually, only 217 pages could be marked as relevant to its content. This significant disproportion was mostly caused by the fact that the portals often have the section of most interesting or most recent articles, what impacted the relevancy estimation based on the word count. On the other hand, these sections increased the probability of discovering the relevant articles. You can see more results in table II. We divided the pages marked as relevant into several categories according to their content and intention. From 217 pages marked as relevant, only 85 had informative character. The others belonged to different categories - list of articles (66 pages), discussions (29 pages), blogs (18 pages), graphical content such as pictures or videos (11 pages), comments (5 pages) or polls (3 pages). We believe that for story tracking, the blogs, discussions and polls are irrelevant because they are mostly reactions on the events. The list of articles is important for the page discovery, but has no informative value for the user. Therefore we propose to use two different heuristics - one for the bee algorithm to collect pages, the other for user to better estimate the page relevancy.

#### C. The Story

If we supposed we had heuristics to filter unwanted content (blogs, discussions etc.) we could build up a story from the found pages (we filtered these pages manually).

We sorted found articles according to the publish date extracted from the page. The algorithm was able to track the story back to the history, the oldest article was published in February 2009.

We divided the real story of presidential elections in Slovakia into five parts and inspected how many pages the bees were able to find (table III).

We can conclude that the algorithm was able to follow the story on the day of the elections. Moreover, it was able to track the story back.

TABLE II  
STATISTICAL RESULTS FROM CASE STUDY

<b>number of found pages</b>	4615
<b>number of pages with <math>q &gt; 0</math></b>	741
<b>number of distinct links</b>	105,044
<b>average number of links per page</b>	88.6

TABLE III  
PARTS OF COVERAGE OF PRESIDENTIAL ELECTIONS IN SLOVAKIA 2009

<b>Story part</b>	<b>Number of found pages</b>
First leg of elections	12
Campaign before the second leg	28
The day of elections (second leg)	30
The results	9
The reactions	6

## VII. CONCLUSION

In this work, we present a simple system for tracking a developing story that is based on a model of a bee hive. We performed a case study that demonstrates the way how our proposed system works. From the case study we can conclude the following:

- the system is able to collect relevant pages
- it can monitor the story being developed during the search
- it can reconstruct the story backwards in time

The case study shows that the system marked many irrelevant pages as relevant. This is not the drawback of the bee algorithm itself, but a conflict between the algorithm and user interests. Algorithm needs non informative pages with the list of articles to be marked as relevant, whereas the user wants only the informative pages to be presented to him. In the future, we propose to use two different heuristics, one for the run of the bee algorithm to collect the set of (possibly) relevant pages, the second one to mine the pages from this set to obtain truly relevant information to the user.

Although the results of the algorithm are not completely satisfying, the story tracking and its aggregation from different data sources is promising.

## ACKNOWLEDGEMENT

This work was partially supported by the Scientific Grant Agency of Republic of Slovakia, grant No. VEGA 1/0508/09.

## REFERENCES

[1] Beekman, M., Gilchrist, AL., Duncan, M., Sumpter, DJT. 2007. What makes a honeybee scout? *Behavioral Ecology and Sociobiology*: 61:985-995.

[2] Ben-Shaul, I., Herscovici, M., Jacovi, M., Maarek, Y. S., Pelleg, D., Shtalhaim, M., Soroka, V., and Ur, S. 1999. Adding support for dynamic and focused search with Fetuccino. In *Proceedings of the Eighth international Conference on World Wide Web (Toronto, Canada)*. P. H. Enslow, Ed. Elsevier North-Holland, New York, NY, 1653-1665.

[3] Biesmeijer, JC., de Vries, H. Exploration and exploitation of food sources by social insect colonies: a revision of the scout-recruit concept. *Behavioral Ecology and Sociobiology*, 49(2-3), pp89-99, 2001

[4] de Bra et al.: *Information Retrieval in Distributed Hypertexts*. Proceedings of RIAO'94, Intelligent Multimedia, Information Retrieval Systems and Management, 1994.

[5] de Vries, H., Biesmeijer, J.C.: *Modelling Collective Foraging by Means of Individual Behaviour Rules in Honey-Bees*, *Behav Ecol Sociobiol* (1998) 109-124.

[6] Dziwiński, P., Rutkowska, D.: *Ant Focused Crawling Algorithm*. In: L. Rutkowski et al. (Eds.): *ICAISC 2008, LNAI 5097*, pp. 1018-1028, 2008.

[7] Gheorghe, M., Holcombe M., Kefalas P.: *Computational models of collective foraging*. *BioSystems* 61 (2001) 133 - 141

[8] Hersovici, M. et al: *The shark-search algorithm - An application: Tailored Web site mapping*. *Proceedings of the seventh international conference on World Wide Web 7*, 1998.

[9] Chakrabarti, S., Berg, M., Dom, B.: *Focused crawling: a new approach to topic-specific Web resource discovery (1999)*. *Computer Networks (Amsterdam, Netherlands: 1999)*

[10] Návrat, P., Kováčik, M., Bou Ezzeddine, A., Rozinajová, V.: *Web Search Engine Working as a Bee Hive*. *Web Intelligence and Agent Systems: An International Journal*. Vol.6 (2008). IOS Press, p. 441-452.

[11] Návrat, P., Jastrzemska, L., Jelínek, T., Ezzeddine Bou, A., Rozinajová V.: *Exploring Social Behaviour of Honey Bees Searching on the Web*, In *Proc.: 2007 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops*, pp. 21 - 25

[12] Menczer, F., Monge, A. E.: *Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study*. *Intelligent Information Agents*, Springer, 1999.

[13] Menczer, F., Belew, R. K.: *Adaptive Information Agents in Distributed Textual Environments*. *Agents*, 1998: 157-164.

[14] Menczer, F., Belew, R. K.: *Adaptive Retrieval Agents: Internalizing Local Context and Scaling up to the Web*. *Machine Learning Journal* 39 (2/3): 203-242, 2000.

[15] Menczer, F., Monge, A. E.: *Scalable Web Search by Adaptive Online Agents: An InfoSpiders Case Study*. *Intelligent Information Agents*, Springer, 1999.

[16] Pant, G., Srinivasan, P., Menczer, F.: *Crawling the Web*. *Web Dynamics*, 2004: 153-178.

[17] Pouliquen, B., Steinberger, R., Deguernel, O.: *Story tracking: linking similar news over time and across languages Coling 2008: Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 49-56 Manchester, August 2008

[18] BeeHive@Work source code: <http://beehiveatwork.sourceforge.net>