An Automated Framework for Mining Reviews from Blogosphere

Arzu Baloglu
Marmara University, Engineering Faculty
Computer Engineering Department
Istanbul, Turkey
E-mail: arzu.baloglu@marmara.edu.tr

Mehmet S. Aktas
Tubitak, The Center Of Research For Advanced
Technologies Of Informatics and Information Security,
Information Technologies Institute, Kocaeli, Turkey
E-mail: mehmet.aktas@bte.tubitak.gov.tr

***Abstract*— As usage of the Blogosphere increases, more and more Internet users have begun to share their experiences and opinions about products or services on the World Wide Web. Web logs (also known as blogs) have thus become an important source of information. In turn, great interest in blog mining has arisen, specifically due to its potential applications, such as collecting opinions regarding products, or reviewing search engine applications for their ability to collect and analyze data. In this study, we introduce an architecture, implementation, and evaluation of a Web log mining application, called the BlogMiner, which extracts and classifies opinions and emotions (or sentiment) from the contents of weblogs.**

Keywords - blog mining, opinion mining, blog crawler, web blog mining

I. INTRODUCTION

The world's biggest library, the World Wide Web, is increasingly populated with data contributed by every Internet user around the world. People share their ideas, interests, emotions, experiences, and knowledge with others, in the form of opinions and reviews, via the Internet every day. Thus, mining opinions on the Web is a rich and important area for research [2].

Sociologists have used many different ways to recognize natural interests, aims, and preferences. In order to collect ideas from people's sharing over the Web, the most efficient way is to mine their Internet diaries, their blogs, which are their own direct, personal accounts of their ideas and opinions. This study introduces a system that is designed to mine ideas to understand the views of a web community.

In the last few years, blogs have emerged as widely known personal Web pages. Blogs began as online diaries. They are designed for regular updating. Each blog consists of a sequence of blog entries. A blog entry consists of a title, a textual content, and the time it was posted. Some blog entries may have comments by the blog readers. Some blogs are dedicated to a particular area of interest such as entertainment or business. Easy-to-use blogging tools have led to an explosion in the number of blogs. Especially with increasing usage of internet, blogging and number of blog pages are growing rapidly. Blog pages have become the most popular means to express one's opinions. By the end of 2008, there were 133 million blogs on the global Internet, as indexed by Technorati [3].

Mining opinions from Web pages involves several challenges. For example, these opinions, or review data, have to be crawled from Web sites and then separated from non-review data [9].

As an experiment, system extracts movie review data from blogs. As a result, we introduce an architecture and we describe the implementation of the system in detail. We also explain a classification of review data.

The organization of this paper is as follows. Section 2 discusses the literature. Sections 3-4 outline the proposed approach and the system architecture. Section 5 addresses the evaluation study. Section 6 concludes the paper with a summary and analysis of results.

II. LITERATURE SURVEY

In recent years, there has been a huge burst of research activity in the areas of sentiment analysis and opinion mining. Earlier studies focused mostly on interpretation of narrative points of view in text [6-11]. The widespread awareness of the research problems in sentiment analysis and opinion mining has increased with the rise of machine learning methods in natural language processing and information retrieval; of the availability of datasets for machine learning algorithms to be trained on (due to the blossoming of the World Wide Web); and, specifically, of the development of review-aggregation Web sites.

Zhongchao Fei et al. [4] describe a sentiment classification application that uses phrase patterns to classify opinions. In this study, at the document classification phase, the authors add tags to certain words in the text, and then match the tags within a sentence with predefined phrase patterns to find the sentiment orientation of the sentence under consideration. Next, they take into account the sentiment orientation of each sentence and classify the text according to the most repeated sentiment.

Jeonghee Yi et al. [5] describe a sentiment miner that extracts sentiment (or opinions) that people express about a subject, such as a company, brand, or product name. In this study, the authors design the sentiment miner with the following challenge in mind: Not only does it try to capture the overall opinion about a topic, but it is also the sentiment regarding individual aspects of the topic, thus capturing essential information of interest. The reason for this is that document-level sentiment classification fails to detect sentiment about individual aspects of the topic. Thus in the author's study, the sentiment miner analyzes grammatical sentence structures and phrases based on natural language processing (NLP) techniques, and detects, for each occurrence of a known topic spot, the sentiment about a

specific topic. With these characteristics, the proposed NLP-based sentiment system [5] achieved high quality results (~90% of accuracy) on various datasets, including online review articles and the general Web pages and news articles. The feature extraction algorithm, proposed by Jeonghee Yi et al. [5], successfully identified topic related feature terms from online review articles, enabling sentiment analysis at finer granularity.

Jian Liu et al. [6] describe an application that completes sentiment classification with review extraction. This approach extracts the review expressions on specific subjects and attaches a sentiment tag and weight to each expression. Then, it calculates the sentiment indicator of each tag by accumulating the weights of all the expressions corresponding to a tag. Next, it uses a classifier to predict the sentiment label of the text. In this study, the authors used on-line documents to test the performance of the proposed application. The experimental documents cover two domains: politics and religion. The experiments within those domains achieve accuracy between 85% and 95%.

Yun-Qing Xia et al. [7] describe a method of opinion mining to help e-learning systems note the users' opinions of the course-wares and e-learning teachers, and thus help improve the services. In this study, the authors develop an opinion mining system for e-learning reviews. The goal of this system is to extract and summarize the opinions and reviews, and determine whether these reviews and opinions are positive or negative. This study divides the whole task into four subtasks: expression identification, opinion determination, content-value pair identification, and sentiment analysis. The authors achieved the following precisions for these subtasks, respectively: 94%, 84.2%, 80.9% and 92.6%.

Qingliang Miao et al. [8] describe a sentiment mining and retrieval system called Amazing. The authors introduce a ranking mechanism, which is different from a general web search engine, since it utilizes the quality of each review rather than the link structures for generating review authorities. In this system, the most important aspect is that the authors incorporate the temporal dimension information into the ranking mechanism, and make use of temporal opinion quality and relevance in ranking review sentences. This study monitors the changing trends of customer reviews in time and visualizes the changing trends of positive and negative opinion respectively. It then generates a visual comparison between positive and negative evaluations of a particular feature in which potential customers are interested. The authors conducted experiments on the sentiment mining and retrieval system using the customer reviews of four kinds of electronic products, including digital cameras, cell phones, laptops, and MP3 players. The evaluation results indicate that the proposed approach achieves a precision of approximately 85%.

Li Zhuang et al. [10] describe a multi-knowledge-based approach that utilizes WordNet for statistical analysis and movie knowledge. WordNet is a large lexical database of English, developed under the direction of George A. Miller [11]. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct concept. The proposed approach, described in [10], breaks down the problem of review mining and summarization into the following subtasks: identifying feature words and opinion words in a sentence; determining the class of feature word and the polarity of the opinion word; identifying the relevant opinion word(s) and then obtaining some valid feature-opinion pairs; and producing a summary using the discovered information. The authors use WordNet to generate a keyword list for finding features and opinions. Grammatical rules between feature words and opinion words are then applied to identify the valid feature-opinion pairs. Finally, the authors re-organize the sentences according to the extracted feature-opinion pairs to generate the summary. The objective of this study is to automatically generate a feature class-based summary for arbitrary online movie reviews. Experimental results show that this method has an average precision of approximately 65%. In addition, with this approach, it is easy to generate a summary with movie-related names as the sub-headlines.

In this study, we extend our previous work described in [1] and propose a project that is most similar to that described by Zhuang et al [10]. Our approach differs from this approach in the way we calculate sentiment orientation of the movie reviews from the blogs. The previous work focused on a constant dataset, while the proposed approach crawls the dataset from the blogs. In turn, this is used to calculate movie scores. We discuss our approach in detail in the next section.

III. APPROACH

A. Overview

In this section, we briefly describe problem definition, the techniques used in this study and what we aim to achieve as a result. This study is categorized into three phases. The first phase is the crawling phase, in which data is gathered from Web logs. The second phase is the analyzing phase, in which the data is parsed, processed and analyzed to extract useful information. The third phase is the visualization phase, in which the information is visualized to better understand the results. More details of the system architecture are explained in the system architecture section (IV).

B. Problem Definition

Web logs are full of un-indexed and unprocessed text that reflects opinions. Many people make choices by taking the suggestions of others into account. For example, one likes to buy a product that is most recommended by people who use that product. Thus, there is a need to crawl and process opinions, so that it can be used in decision-making processes of potential Web review applications.

C. Solution

In this study, we propose a blog mining system that will extract movie comments from Web logs and that will show Web log users what other people think about a particular

movie. Figure 1 shows the overall process model of the proposed system. As illustrated in this Figure 1, the blog mining process consists of following three main steps: Web crawling, sentiment analysis, and visualization.

Web crawling: A Web crawler (also known as a Web spider, or Web robot) is a program or automated script that browses the World Wide Web in a methodical, automated manner. A Web crawler is a type of software agent that takes a list of URLs, called seeds, to visit as input. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to a list of URLs, called the crawl frontier, to visit. URLs from the frontier are recursively visited according to a set of policies. The process of Web crawling is also known as spidering. Many sites, and search engines in particular, use spidering as a means of providing up-to-date data. Web crawlers (or spiders) are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for spam) or gathering text content.

In this study, we utilized two open source projects, OpenWebSpider [16] and Arachnode [21], for crawling the Web logs and collecting data for sentiment analysis.

Sentiment analysis: Sentiment analysis has three main tasks: determining subjectivity, determining sentiment orientation, and determining the strength of the sentiment orientation.

identifies whether the keyword has an adverb, which changes the degree of subjectivity. For determining the sentiment orientation the algorithm calculates the cumulative sentiment score for the review. If a keyword under consideration is found in the database, then the algorithm calculates the score.

Visualization: We utilize the Zed Graph [15] for visualization to present our findings. The Zed Graph provides an ASP web-accessible control for creating 2D line, bar, and pie graphs of arbitrary datasets. It is maintained as an open-source development project. We presented the results on the project website over a shared database.

IV. System Architecture

The proposed system architecture consists of several components: Blog Crawler, Sentiment Analyzer, and Web Usage Interfaces.

A. *Blog Crawler*

One of the most important parts of the proposed system is the blog crawler. The crawler needs to analyze as much data as possible to provide accurate results. If the analysis has not been conducted with enough data, the results will only indicate the opinions of a restricted group of people. Although one needs to crawl as many blogs as possible to obtain good results, the blogosphere contains huge amounts of data. The storage capacity is limited, and limitations also exist related to the computation and memory capabilities necessary to crawl all of the blogosphere.

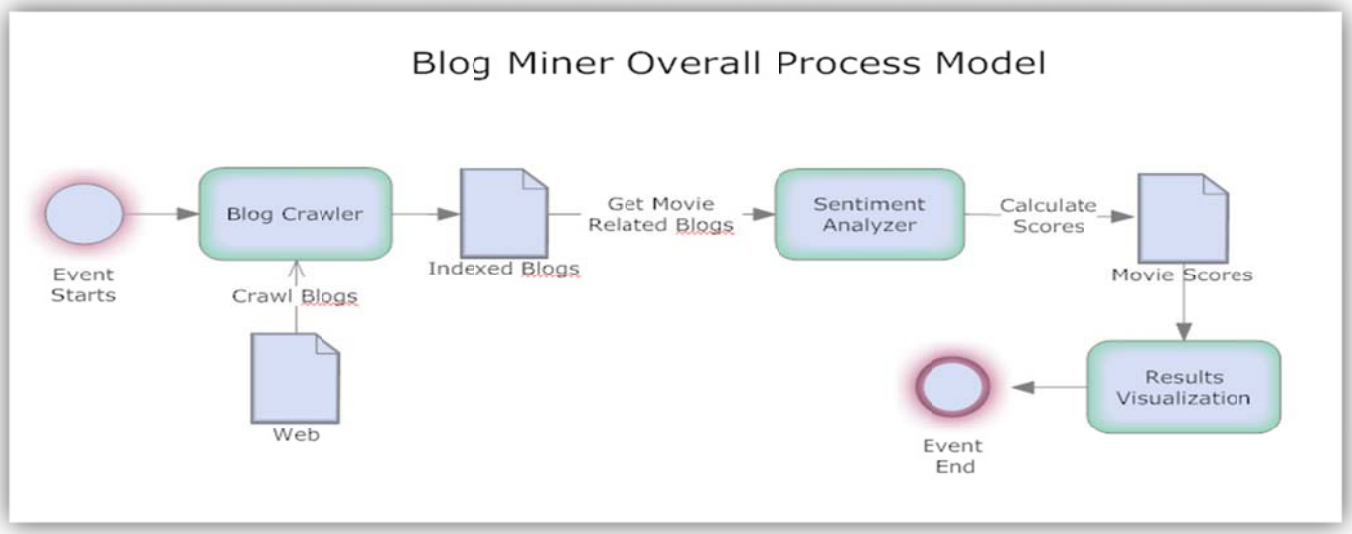


Figure 1 Blog Miner Overall Process Model

In this study, we use an unsupervised approach to sentiment analysis. For determining subjectivity, we use a keyword algorithm, which searches the pre-defined keywords in the text and then calculates their sentiment scores. For determining sentiment orientation, the algorithm

In this study, when calculating the general opinions about a movie, we were only able to crawl part of the blogosphere. We can assume that when the computation capabilities are improved, and the crawled area of the blogosphere is

increased, the proposed application will produce better results.

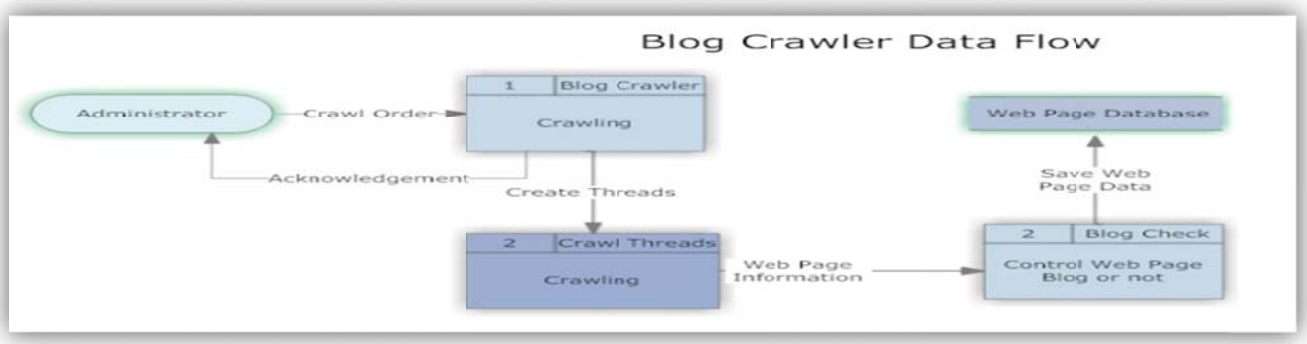


Figure 2 Blog Crawler Data Flow

We used Arachnode.Net to crawl the Web logs. Arachnode.net is an open source Web crawler for downloading, indexing, and storing Internet content including e-mail addresses, files, hyperlinks, images, and Web pages. Arachnode.net is written in C# and uses SQL Server 2005.

Arahnode.net uses the Lucene.Net library for indexing and searching. Arachnode.Net is selected because it is very customizable and well written. We start with seed lists like www.blogpulse.com and www.technorati.com, because these Web sites contain many links to Web logs. In turn, this improves the crawling performance. Figure 2 shows the data flow in the crawling process, while Figure 3 shows the main working process of the crawler.

As illustrated in Figure 3, the Web crawler starts by parsing a set of links that point to blog pages. The crawler then parses those pages for new links, and so on, recursively. After the new links are extracted, the system checks if they point to blog pages and inserts them into a queue of links to be processed by the crawler. The crawler resides on a single machine and sends HTTP requests for documents to other machines on the Internet, just as a web browser does when the user clicks on links. If the page is already fetched and resides in the cache, the crawler omits the link pointing to this page. All the crawler really does is to automate the process of following links for blog pages.

B. Sentiment Analyzer

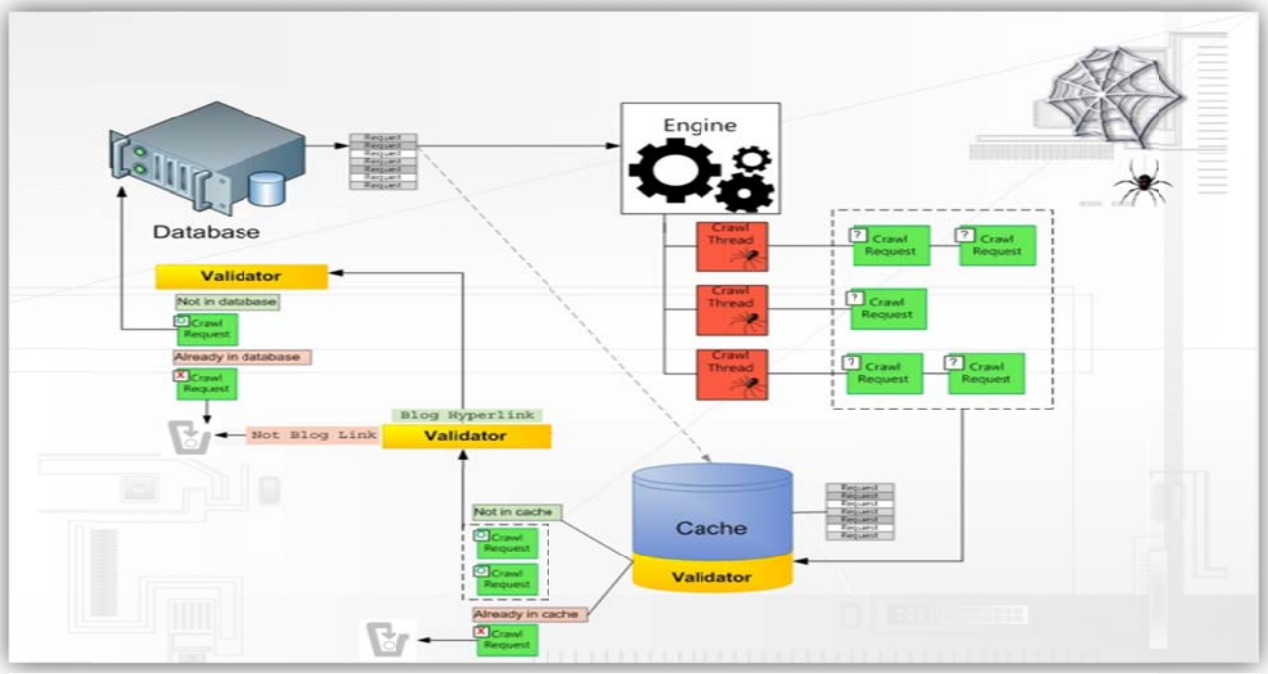
The sentiment analyzer is a crucial component of the proposed system. If the analyzer finds a pre-defined keyword in a sentence of a given blog page for a specific movie, it looks for the sentiment words (such as an adjective or an adverb) that may be associated with that keyword. It calculates the sentiment scores for a movie for different keywords by mining the comments from blog pages. In order to calculate the sentiment scores, the analyzer first selects the blog pages that contain comments about a specific movie. Then, it parses each blog text and processes it in order to calculate sentiment scores for different keywords related to the movie under consideration.

The sentiment analyzer utilizes the aforementioned keyword algorithm in order to calculate sentiment scores. In this algorithm, the sentiment analyzer processes every sentence of a blog page for keywords such as “Screenplay,” “Director” and “Producer” that are related to the movie domain.

The analyzer utilizes the SentiWordNet [12] to obtain the sentiment scores. The SentiWordNet is a lexical resource, where each WordNet [11] synset *s* is associated with three numerical scores Obj(*s*), Pos(*s*) and Neg(*s*), describing how objective, positive, and negative the terms contained in the synsets are. Figure 3 shows some adjectives and their scores according to the SentiWordNet.

If it finds a sentiment word, it obtains its score from SentiWord. It uses the obtained score as the keyword’s score and adds that to the total sentiment score of the blog page.

If the analyzer finds an adjective in a sentence of a given blog for a specific movie, it also looks for an adverb that modifies the degree of the adjective. Here, the adverbs are separated into two main categories, degree-adverbs and reversing-adverbs. If the analyzer finds a degree-adverb such as “less” or “more” in front of an adjective, then it multiplies the adjective’s score by the degree-adverb’s score and uses the result as the keyword’s score. If the analyzer finds a reversing-adverb such as “not” in front of an adjective, it simply reverses the score of that adjective and uses the result as keyword’s score.



id	wordid	type	posscore	negscore	objectivity	word
241669	800506	a	0.25	0.375	0.375	trenchant
241670	800669	a	0.125	0.25	0.625	impelling
241672	80096	a	0.375	0	0.625	accepting
241673	800997	a	0	0	1	rough-and-ready
241678	801906	a	0	0.875	0.125	ineffectual
241679	802074	a	0	0.625	0.375	effortful
241681	802754	a	0	0.625	0.375	difficult
241684	80310	a	0.125	0.625	0.25	rejective
241685	803227	a	0.125	0.625	0.25	labored
241686	803386	a	0	0.5	0.5	labor-intensive
241687	803568	a	0	0	1	leaden
241690	804036	a	0.25	0.5	0.25	effortless
241692	804388	a	0.5	0	0.5	unforced
241693	804585	a	0.5	0	0.5	efficacious
241694	80477	a	0	0.125	0.875	repudiative
241696	805086	a	0.25	0.125	0.625	inefficacious
241697	805276	a	0.375	0.375	0.25	efficient
241698	805616	a	0.25	0	0.75	businesslike
241699	805760	a	0	0	1	cost-efficient
241700	805869	a	0.125	0	0.875	economical

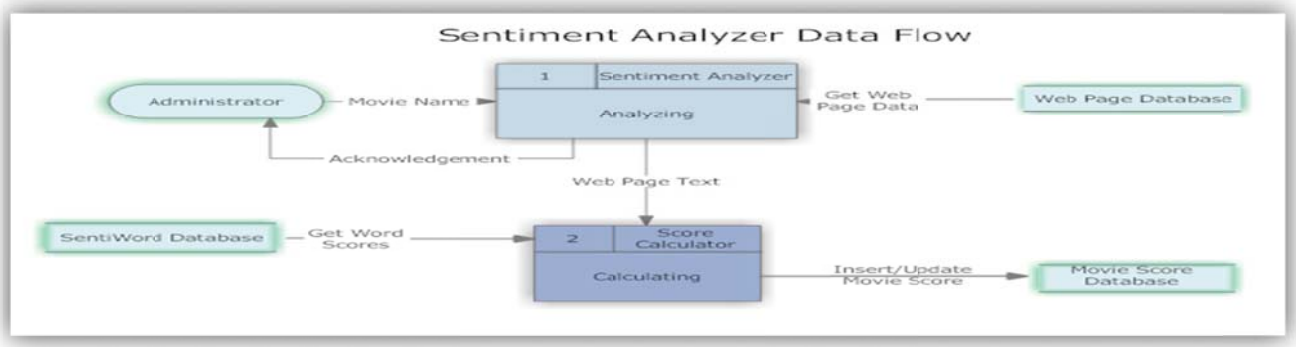


Figure 5 Sentiment Analyzer Data Flow

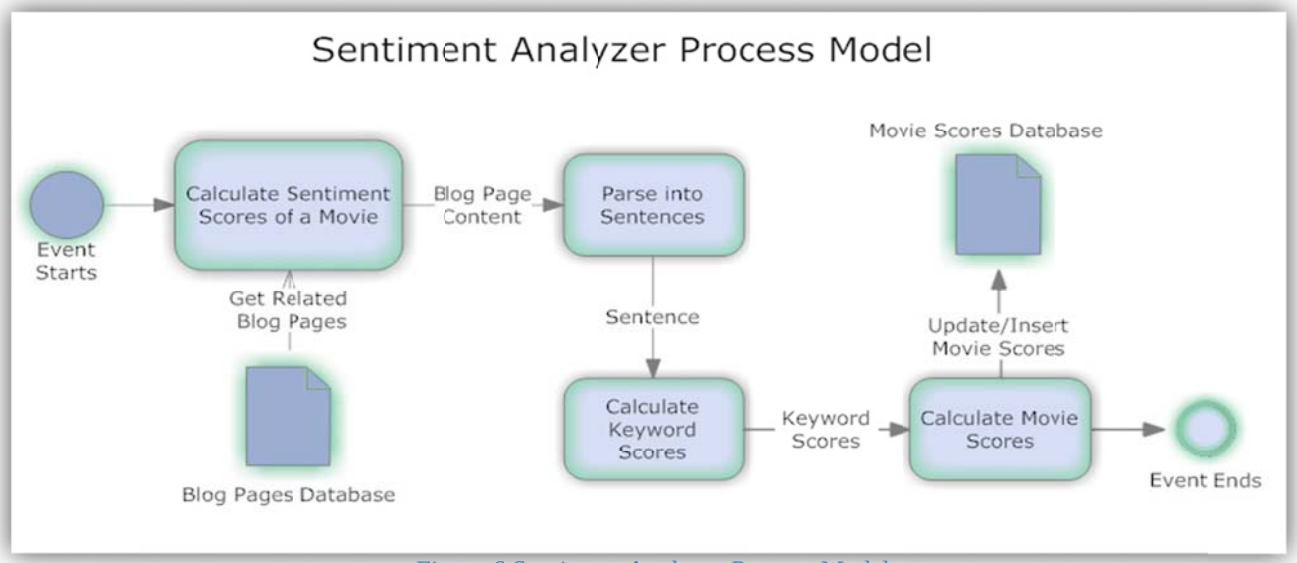


Figure 6 Sentiment Analyzer Process Model

In this way, the analyzer calculates the cumulative sentiment scores for all related blog pages for different pre-defined keywords and takes the average of these scores. In the end, the analyzer finds a sentiment score corresponding to each pre-defined keyword by mining the blogs for each particular movie. Figure 5 shows the data flow in the sentiment analyzer.

The reviews and comments in blogs may contain spelling errors, and these errors will decrease the accuracy of the application. To overcome this challenge, NetSpell [16] is used as a spelling library in our score calculation methodology.

The SentiWordNet database contains the stem of the words. In turn, this may affect the calculation of the sentiment scores and decrease accuracy of the application. Thus, in order to discover the sentiment score of a word, the analyzer must search its stem within the SentiWordNet. To overcome this problem, the analyzer utilizes the Porter Stemmer [14] to get the stem of a word. These text and word modifications improve the proposed application's accuracy. Figure 6 shows the process model of the sentiment analyzer.

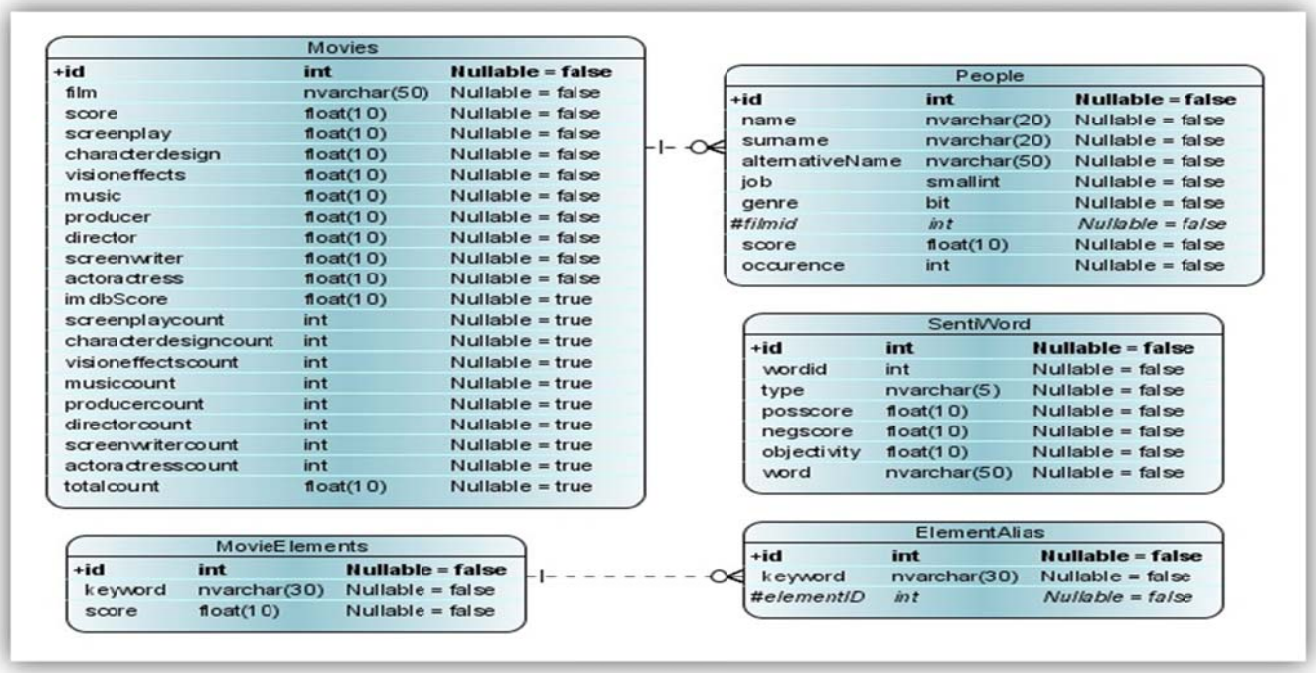


Figure 7 Blog Miner ER Diagram

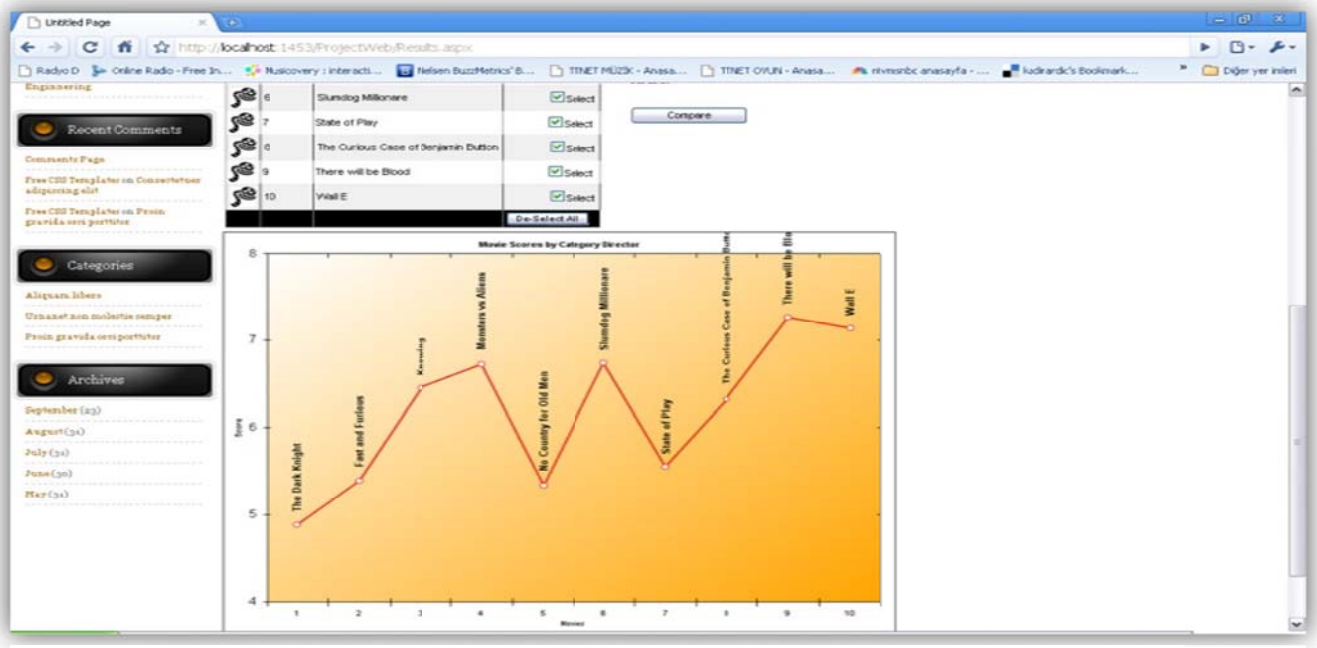


Figure 7 shows the Entity Relationship Diagram of the proposed application. Note that this diagram does not include the Arachnode.Net database, which is used to store blog pages. The database diagram of the Arachnode.Net is available at [12]. In Figure 7, the "Movies" table is used to store the score results of each movie under investigation. The "People" table is used to store all related information about the people involved in making movies, such as actors, actresses, and directors. The "SentiWord" table stores the sentiment dictionary, which was obtained from SentiWordNet [11]. The "Movie Elements" table stores the nine keyword categories from the movie domain, and the "Element Alias" table stores the keywords associated with these categories.

C. Web User Interfaces

We developed a Web user interface, as illustrated in Figure 6, to present the project evaluation and to give information about ongoing research. The web interface is used for two functions: The first category is the selection. There are two types of selection options. First is the selection of movies. Here, the system lets the user select a movie and then shows the sentiment score results corresponding to nine different keyword categories. Second is the selection of keyword categories. Here, the system lets the user specify only one category and shows the sentiment scores of different movies under the selected keyword category. In addition, the system also lets users select the movies they want to sketch and the category under which they want to do the analysis, and then shows the results in a graph.

The second category is the graphs. The system utilizes dynamic charts that are created each time users specify a selection as illustrated in Figure 7. Here, we utilize Zed Graph [15], which is an open-source library, written in C#, for creating 2D line and bar graphs of arbitrary datasets. This library provides a high degree of flexibility, i.e., almost every aspect of the graph can be user-modified.

Zed Graphs has two different libraries that can be used for creating Windows-based applications and Web-based applications. In this study, we use only some parts of the Zed Graph libraries to create a Web-based BlogMiner application.

V. EXPERIMENTS AND RESULTS

To evaluate the performance of the proposed application, we used reviews about several movies from The Internet Movie Database (IMDD) Web site [18] as the data set. For our analysis, we simply chose recent movies, since we want to analyze as many user comments as possible. Our assumption is that recent movies will attract more user comments, as they may have a larger audience.

Thus, we chose following 10 movies from the IMDB: "The Fast and Furious," "Monsters vs. Aliens," "State of Play," "Knowing," "The Dark Knight," "Wall-E," "Slumdog Millionaire," "No Country for Old Men," "There Will be Blood" and "The Curious Case of Benjamin Button." For each movie, approximately 10 review pages are crawled by the Blog Crawler. In turn, this created approximately 1000 user reviews in total. These reviews are used for experiments to calculate the accuracy of the application. For the pre-defined keywords, we used the names of the three most important roles for each movie: actor/actress, director, and screenwriter. We include the names of these roles in the database in order to catch comments about actors, actresses, directors, and screenwriters. We refer the readers to Ardic and Enez [19] for extensive discussion on implementation and experiments.

We present our experimental study by showing the steps of the BlogMiner application for processing the raw data and calculating sentiment scores. Thus, the following sample user-review is chosen from IMDB to illustrate the steps.

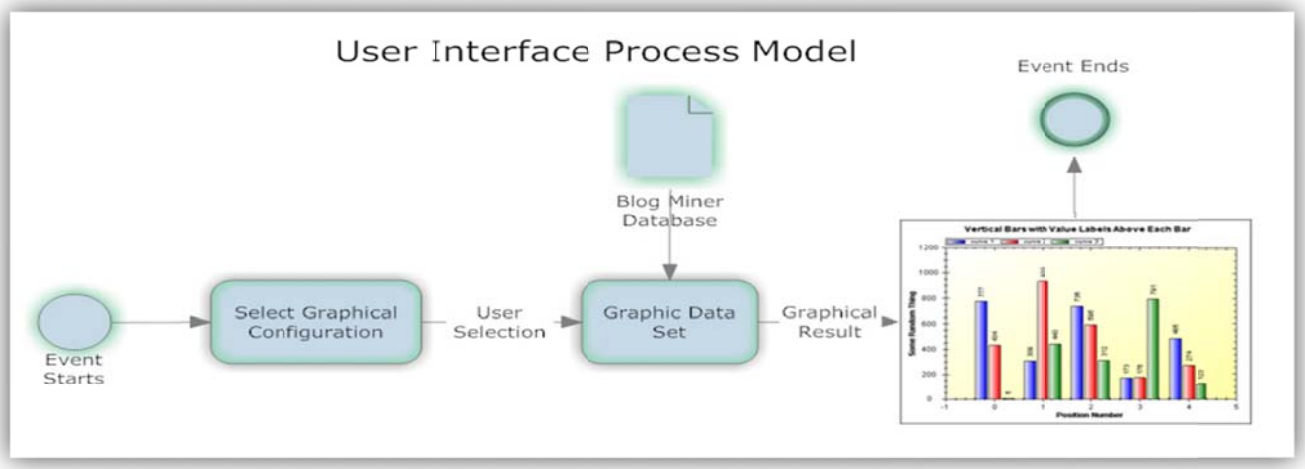


Figure 9 Web User Interface Process Model

Sample Review: "I thought it wouldn't be as good as it was, because thousands of people and reviews said it would suck! It was great, but what it missed was that it needed to be at-least an hour longer, because it missed a-little bit, but it still rocked! I loved it! I thought it was funny, and as did the person next to me, when John says: "I'll be back!"".

First we split the text into sentences: In this step, the BlogMiner simply breaks down the text into sentences and makes the sentiment analysis at sentence level. Below, we illustrate this process as applied to the sample review after step 1.

~1~ I thought it wouldn't be as good as it was, because thousands of people and reviews said it would suck! **~1~**
~2~ It was great, but what it missed was that it needed to be at-least an hour longer, because it missed a-little bit, but it still rocked! **~2~**
~3~ I loved it! **~3~**
~4~ I thought it was funny, and as did the person next to me, when John says: "I'll be back!"". **~4~**

Second, we tag the words in each sentence by their type. In this step, appropriate tags are added to the words to be able to understand their meanings more accurately. Figure 8 shows the tags that have been used and the meanings of these tags. The text below is the sample review after step 2.

I/**PRP** thought/**VBD** it/**PRP** would/**MD** not/**RB** be/**VB** as/**RB** good/**JJ** as/**IN** it/**PRP** was/**VBD ,/,** because/**IN** thousands/**NNS** of/**IN** people/**NNS** and/**CC** reviews/**NNS** said/**VBD** it/**PRP** would/**MD** suck/**VB !/.**

It/**PRP** was/**VBD** great/**JJ ,/,** but/**CC** what/**WP** it/**PRP** missed/**VBD** was/**VBD** that/**IN** it/**PRP** needed/**VBD** to/**TO** be/**VB** at-least/**JJ** an/**DT** hour/**NN** longer/**RB ,/,** because/**IN** it/**PRP** missed/**VBD** a-little/**JJ** bit/**NN ,/,** but/**CC** it/**PRP** still/**RB** rocked/**VBD** !/.

I/**PRP** loved/**VBD** it/**PRP !/.**

I/**PRP** thought/**VBD** it/**PRP** was/**VBD** funny/**JJ, /,** and/**CC** as/**RB** di/**VBD** the/**DT** person/**NN** next/**JJ** to/**TO** me/**PRP, /,** when/**WRB** John/**NNP** says/**VBZ :/:** "/`` I/**PRP** will/**MD** be/**VB** back/**RB !/. ”/NN. /.**

Third, we score the text using the keyword algorithm and calculate the scores. In this step, the system calculates the sentiment score for keywords and finds the accumulated scores for each sentence. Below, we illustrate the output of the sample review after step 3. The results of the experiments are illustrated in Figure 9 .

“I/**PRP** thought/**VBD** it/**PRP** would/**MD** not/**RB**<-1> be/**VB** as/**RB** good/**JJ**<0.844> as/**IN** it/**PRP** was/**VBD ,/,** because/**IN** thousands/**NNS** of/**IN** people/**NNS** and/**CC** reviews/**NNS** said/**VBD** it/**PRP** would/**MD** suck/**VB !/.**
(sentence score = -0.844)

It/**PRP** was/**VBD** great/**JJ**<0.344> **,/,** but/**CC** what/**WP** it/**PRP** missed/**VBD** was/**VBD** that/**IN** it/**PRP** needed/**VBD**<-0.140625> to/**TO** be/**VB** at-least/**JJ** an/**DT** hour/**NN** longer/**RB ,/,** because/**IN** it/**PRP** missed/**VBD** a-little/**JJ** bit/**NN ,/,** but/**CC** it/**PRP** still/**RB**<-0.171> rocked/**VBD** !/.
(sentence score = 0.0104)

I/**PRP** loved/**VBD**<0.375> it/**PRP !/.**
(sentence score = 0.375)

I/**PRP** thought/**VBD** it/**PRP** was/**VBD** funny/**JJ**<-0.515> **,/,** and/**CC** as/**RB** did/**VBD** the/**DT** person/**NN** next/**JJ** to/**TO** me/**PRP ,/,** when/**WRB** John/**NNP** says/**VBZ :/:** "/`` I/**PRP** will/**MD** be/**VB** back/**RB !/. ”/NN ./.**
(sentence score = -0.515)

As can be seen in this figure, the producer and screenwriter columns include rows with a score of 5.25. These scores are default values because no keywords were found for these movies.

The results of the experiment have been compared with each movie's IMDB score. On the IMDB page of each movie, the movie's general scores are listed. Thus, we can compare the IMDB score against the keyword algorithm's score.

For the producer and screenwriter categories, not enough comments were found to calculate a realistic score. As a result, most of the producer and screenwriter score columns are given the default value. When the results are compared against the IMDB scores, we observe a similar behavior. A movie with a low IMDB score also gets a low score in the proposed application. Similarly, a movie with high IMDB score gets a high score in the proposed application. We also observe two exceptions to this behavior. For example, the movies "Fast and Furious" and "State of Play" received high scores in our application; however, their IMDB scores are in a lower position than the proposed application calculated. We conclude that in the IMDB database, the comments and the score of the movie may not always be matched correctly.

VI. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Opinion mining is an important area of investigation. As Web 2.0 applications produce an enormous collection of meaningful information, mining such information has become an important task. In this study, we introduced an opinion mining application that is created for calculating movie scores from blog pages.

film	score	screenplay	characterdesign	visioneffe...	music	producer	director	screenwri...	actoractress	imdbScore	keywordscore
The Dark Knight	7.77	5.14	4.78	4.97	5.46	6.88	4.88	6.23	5.31	9	7.64
Fast and Furious	8.05	5.05	5.3	5.47	6.84	5.25	5.39	5.25	4.94	6.9	7.73
Knowing	7.6	5.26	7.5	4.98	2.81	1.11	6.45	5.25	5.84	6.8	7.41
Monsters vs Aliens	7.65	5	5.08	5.82	4.95	5.25	6.72	1.56	6.35	7	7.73
No Country for Old Men	7.69	5.37	5.47	6.12	5.63	5.25	5.34	5.25	5.55	8.3	7.75
Slumdog Millionare	7.89	5.16	5.88	5.47	5.39	5.25	6.74	5.25	5.6	8.5	7.88
State of Play	8.28	4.98	5.25	6.68	5.49	5.25	5.55	5.25	5.35	7.9	7.83
The Curious Case of Be...	7.64	5.09	3.57	4.78	5.38	5.25	6.32	5.25	5.3	8.2	7.56
There will be Blood	7.78	5.03	5.31	5.97	4.38	5.25	7.26	5.25	6.09	8.3	7.59
Wall E	8.14	5.43	5.92	5.53	5.31	5.25	7.14	4.32	6.38	8.6	7.82

Figure 10 Experiment Results

CC	Coordinating conjunction	RP	Particle
CD	Cardinal number	SYM	Symbol
DT	Determiner	TO	to
EX	Existential there	UH	Interjection
FW	Foreign word	VB	Verb, base form
IN	Preposition/subordinate conjunction	VBD	Verb, past tense
JJ	Adjective	VBG	Verb, gerund/present participle
JJR	Adjective, comparative	VBN	Verb, past participle
JJS	Adjective, superlative	VBP	Verb, non-3rd ps. sing. present
LS	List item marker	VBZ	Verb, 3rd ps. sing. present
MD	Modal	WDT	wh-determiner
NN	Noun, singular or mass	WP	wh-pronoun
NNP	Proper noun, singular	WP$	Possessive wh-pronoun
NNPS	Proper noun, plural	WRB	wh-adverb
NNS	Noun, plural	``	Left open double quote
PDT	Predeterminer	,	Comma
POS	Possessive ending	''	Right close double quote
PRP	Personal pronoun	.	Sentence-final punctuation
PRP$	Possessive pronoun	:	Colon, semi-colon
RB	Adverb	$	Dollar sign
RBR	Adverb, comparative	#	Pound sign
RBS	Adverb, superlative	-LRB-	Left parenthesis *
		-RRB-	Right parenthesis *

Figure 11 Word Tags

Experimental results show that the proposed application produces accurate results close to IMDB result values. With this study, we introduced an unsupervised approach for sentiment analysis.

For future study, we want to further improve this application and investigate how clustering and self-organization methodologies can be used to improve the accuracy in the results. We will further improve the software so that the users are able to add their own keywords at runtime. We will also investigate the scalability of this approach by investigating the system performance under an increasing number of keywords.

Acknowledgement: We thank Kadir Ardic and Onur Enez for their contribution to the research presented in this paper. We also thank the Department of Computer Engineering in Marmara University for giving us permission to commence this study and to do the necessary research work by utilizing departmental computer facilities.

REFERENCES

[1] Baloglu, Arzu, Aktas, Mehmet, Mining Movie Reviews from Web Blogs: An approach to Automatic Review Mining, Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference, Barcelona, Spain, 9-15 May 2010

[2] Bing Liu, Web Data Mining - Exploring Hyperlinks, Contents and Usage Data, Text Book, , Springer, December, 2006

[3] Technorati Web Site is available at http://technorati.com, last accesed October 2009

[4] Zhongchao Fei, et al., Sentiment Classification Using Phrase Patterns Proceedings of the Fourth International

Conference on Computer and Information Technology (CIT'04), 2004.

[5] Jeonghee Yi, et al., Sentiment Mining in WebFountain, Proceedings of the 21st International Conference on Data Engineering (ICDE 2005), 2005

[6] Jian Liu, et al., Super Parsing: Sentiment Classification with Review Extraction, Proceedings of the Fifth International Conference on Computer and Information Technology (CIT'05), 2005.

[7] Yun-Qing Xia, et al., The Unified collocation Framework for Opinion Mining, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007.

[8] Qingliang Miao, et al., AMAZING: A sentiment mining and retrieval system, Expert Systems with Applications (2008) doi:10.1016/j.eswa.2008.09.035.

[9] Qiang Ye, et al., Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert Systems with Applications (2008) doi:10.1016/j.eswa.2008.07.035.

[10] Li Zhuang, et al., Movie review mining and summarization, Proceedings of the 15th ACM international conference on Information and knowledge management, 2006.

[11] WordNet Web site is available at http://wordnet.princeton.edu, Access Date: October 2009.

[12] Andrea Esuli, et al., SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining, The fifth international conference on Language Resources and Evaluation, LREC 2006

[13] Arachnode.Net Web site is available at http://arachnode.net/media/g/databasediagrams/-default.aspx, last accessed October, 2009

[14] Porter Stemmer Web site is available at http://tartarus.org-/~martin/PorterStemmer, Access data: October 2009

[15] Zed Graphs Web site is available at http://zedgraph.org-/wiki/index.php?title=Main_Page, Access date: October 2009

[16] NetSpell Web site is available at http://sourceforge.net-/projects/netspell, Access date: October 2009

[17] OpenWebSpider Web site is available at http://www.openwebspider.org, Access date: October 2009

[18] The Internet Movie Database (IMDB) Web site is available at http://www.imdb.com, Access date: October 2009

[19] Kadir Ardic, Onur Enez, Blog Mining, Undergraduate graduation thesis is available at http://www.scribd.com/doc/-16191423/Web-Blog-Miner-Licence-Thesis, last accessed October 2009