A Model for Opinion Ranking

Allan D. S. Lima, Jaime S. Sichman

Laboratório de Técnicas Inteligentes (LTI)
Escola Politécnica da USP
Avenida Prof. Luciano Gualberto, travessa 3 nº 158 - CEP - 05508-970 - São Paulo – SP
adsl@usp.br, jaime.sichman@poli.usp.br

Abstract. *Day after day we have been noticing the growth in the number of opinions about any subject over the Internet. Consequently it's becoming hard for people to know what others think about a subject. In order to proper handle this problem, a research field called opinion mining was created. One of its objectives is to classify the relevance of an opinion. Focusing in this point, we present in this paper a model to rank opinions. Our model is based in the composition of a ranking function by combining many concepts. Some of them are related to social aspects of the user who is searching for opinions. Also we discuss how to find the best combination of our parameters and how our model is different for projects where opinions were ranked before.*

1. Introduction

When someone wants to find opinions about a subject (product, theme, person, etc.) how can he have access to third party opinions? Before the Internet, when someone wanted to discover opinions about a subject he could ask for people in his social circle like his parents, relatives, friends, etc. Furthermore, other sources like television, magazines and newspapers were frequently consulted as well. However nowadays, it's possible to search over the Internet for opinions about basically every subject. By using tools like websites, forums, blogs and search engines it is possible to everyone to express or to access information, as well as opinions. Ironically, as a side effect, this also brought an information overload, making hard to find the opinions that are really relevant among all those available.

For example, how someone interested in buying an electronic game called Just Dance can find opinions about this product? One of the most common ways is to access a search engine (like Google) and search for "just dance review". This query will give the user more than 60 million pages and probably each of them has many opinions in its content! Hence, the problem we will discuss in this paper can be described as: *among the data available at the Internet, how to find the opinions that are relevant to someone?* We believe that the best way to solve this problem is by proposing a model to rank opinions capable of give personalized results, i.e., respecting the fact that every user can have its own way to define an opinion as relevant. In other words, our main goal is to build a model that can generate a personalized opinion list depending on who is searching the opinion. In order to deal with this problem, we are going to introduce a unique way to use Opinion Mining [Liu 2012] together with other concepts like Social Search [Morris at al 2010a], a concept that even Google have been applying on its document ranking functions [Google 2012]. In order to achieve our objective, we have been developing an opinion ranking model composed by: (1) a set of concepts that can evaluate the relevance of an opinion; (2) an abstract (domain independent) ranking function for opinions able of use those concepts as parameters; (3) a way to discover how to proper combine those concepts in a scoring function.

So far we have defined seven concepts as the parameters of our ranking function: (1) Information Retrieval [Manning al at 2008]; (2) Memes [Dawkins 2006]; (3) author's experience; (4) how a user sees an author as relevant or not; (5) how all users see an author as relevant or not; (6) the similarities between a user and an author; (7) the distance between an user and an author in a relationship network (like a graph). Also it's important to highlight that parameters like (4) allow our model to give different scores to the same opinion depending of the user is searching for it. We made this possible because the same opinion can relevant for a user while it can irrelevant to other.

This paper describes the current state of our work. We start from an introduction to opinion mining and other basic concepts necessary to proper understand our model, in section 2. Then we review the previous work related to our objective in section 3. In section 4 we describe our model, while in section 5 show how our model handles opinions differently for others projects. Finally, in section 6 we made our final considerations and present our plans for future work.

2. Basic Concepts

2.1. Opinion Mining

Textual information can be categorized basically in two types: (1) facts and (2) opinions. Facts are objective declarations about entities or events while opinions are declarations reflecting subjective fillings of people or perceptions about entities or events [Liu 2012]. Since the popularization of the Internet, especially regarding to tools like blogs, the way people express and have access to opinions have been changing. For example, when someone wants to make his mind about a polemic subject, he can use not only the traditional ways (friends, relatives, television, newspapers or magazines) but also the whole information present at the Internet [Liu 2010] as well.

Information Retrieval [Manning et al 2008] is the computer science research field responsible of mining information from the Internet. However, research in this field in most cases does not handle facts and opinions differently, both are considered as information. Consequently those techniques aren't able to proper help users demanding for opinions. This is one of the reasons why researchers created a new research field called Opinion Mining that can be defined as follow [Liu 2012]:

Definition 1 – Opinion Mining: *Given a set of documents D containing opinions (or sentiments) about a subject, opinion mining is considered to be a set of tasks of extraction and identification of features and components from the subject addressed by each document d in D, and then defining if these comments are positive, negative or neutral.*

As an example to differentiate opinion mining from information retrieval, let us analyze an hypothetical piece information about a camera: "it has a good image resolution, but its zoom isn't as good as other cameras already in the market". It's possible to notice the existence of two different opinions in the text, the first one is positive and the second one is negative. Moreover the first opinion is directly related to the image resolution. Meanwhile the second refers to the zoom feature by a comparison with other cameras. While information retrieval will handle that sentence as a text document piece, opinion mining will try to identify and eventually index opinion aspects presents on it. This simple example was responsible to intuitively introduce some of the opinion mining targets. But there are many other research objectives related to this research field, among them we can highlight [Liu 2010]:

- **Problem formalization:** mathematically handle problems related to opinion mining through formalizations of its definitions, limitations, constraints or objectives;
- **Sentiment and subjectivity analysis:** given a document, identify if it has opinions, and if so, classify them as positive, negative or neutral;
- **Characteristic oriented sentiment analysis:** discover the features that were referred in each opinion. For example, for a computer it's possible to have opinions about features like its *processor* or its *hard drive*.
- **Sentiment analysis in comparative sentences:** usually an author expresses his opinion about a theme by comparisons or metaphors. For example, "the notebook battery lasts as long as every ordinary battery";
- **Search engine and information retrieval:** systems where the user is able to make queries about a theme and receives as result a list of opinions or documents (containing opinions);
- **Spam detection and opinion utility:** When it comes to opinion mining spam is also a problem; for example, it's possible that an author paid by the product manufacturer could make product reviews. Also the utility of an opinion is a score that represents its importance degree so it can be used to create ranking documents functions.

2.2. Information Retrieval

Information Retrieval (IR) is seen like the task of search for data (usually documents) in non-structured documents (usually texts) in order to satisfy a specific information need in a big collection of data (usually stored in computers) [Manning et at 2008]. In order to proper understand this definition it's necessary to understand the difference among structured and non-structured data before. A dataset is structured when its information is well delimited by fields that have a semantic value that makes it easy to parse by an algorithm. For example, table lines in a relational database are considered as structured data. However, a dataset is considered as non-structured when there is no distinction among its different parts, which makes it hard to parse by an algorithm. For example, a text in an Internet blog is considered to be a set of non-structured dada.

Because of the big collections of data, the results of a query in an IR system can have millions of documents, what exceeds the human capacity of reading them in a short time interval. Hence, IR provides many strategies in order to score documents and consequently generate an ordered list of them. One of the most simple and popular ways to rank documents uses the frequency of terms in the documents. The *tf-idf* score function is composed by two concepts the (1) *tf*: term frequency and the (2) *idf*: inverse document frequency [Manning at al 2008]. The term frequency is the number of occurrences of a term in a given document. But the inverse document frequency is computed according to Equation 1. The *idf* score is logarithm of the number of documents (N) divided by the number occurrences of that a term in the all documents in a collection (*df*).

$$idf_t = log\frac{N}{df_t}$$

Equation 1.Inverse document frequency.

The final value of the *tf-idf* score is archived by the multiplication of the both (*tf x idf*). But since this is value of a single term, we need to sum the *tf-idf* score for each term in a query in order to compute the score of a document.

Ranks generated by score functions like *tf-idf* have to be evaluated, this why in information retrieval there are many metrics to measure how an information retrieval system is good. In this section we are going to cover four of them [Manning et al 2008]: (1) precision; (2) coverage; (3) F Measure; (4) Means Average Precision;

Precision, Equation 2, is computed by the division of the number of relevant items retrieved by the number of retrieved items. In other words precision represents the conditional probability of a relevant item, given that it was retrieved. An important variation of this metric is the K-Precision (also known as P@K) where only the first K retrieved items are considered.

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(items\ retrieved)} = P(relevant|retrieved)$$

Equation 2. Precision.

Coverage, Equation 3, is computed by the division between the number of relevant items retrieved and the number of items retrieved. In other words, it's the conditional probability that an item be retrieved given that it is relevant.

$$Coverage = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = P(retrieved|relevant)$$

Equation 3. Coverage.

In order to combine both (precision and coverage) there is a metric called F Measure. It unites both by the harmonic mean, Equation 4, where α represent the weight of the combination. For example, if α > 0.5 then it means that precision is more important than coverage.

$$FMeasure = \frac{1}{\propto\frac{1}{P} + (1-\propto)\frac{1}{C}}$$

Equation 4. F Measure.

Another important metric used to evaluate information retrieval systems is the Means Average Precision. Given that during the evaluation of a system many query are submitted to it, then this concept is computed by the average precision for the first K documents in each query during the tests. Hence, given $Q = \{q_0,\dots q_j,\dots q_n\}$ as the set of all test queries; given $D = \{d_1, \dots, d_{mj}\}$ as the relevant documents for a query q_j; and given R_{jk} as the set retrieved documents at the first k documents; Equation 5 shows how to compute the Means Average Precision.

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk})$$

Equation 5. MPA.

2.3. Social Search

Social Search is been expected to be the future of search engines, researchers [Evans and Chi 2010], information retrieval systems [Horowitz and Kamvar 2010; ChaCha 2011], and companies like Google [Sherrets 2008] and Microsoft [Morris et al 2010a; Morris et al 2010b] have been studying about how social components can help a user to fulfill its information needs. Experiments developed by Microsoft researchers showed that people usually have more confidence in answers from its friends. Also they have discovered that people frequently believe that social networks are better than search engines when answering subjective questions like products recommendations [Morris et al. 2010a]. Moreover, the experience of information searching could be improved by the integration of search engines and social resources [Morris et al 2010b].

It's possible to understand the meaning of social search by two complementary definitions:

Definition 2 – Social Search by Morris et al: *The term social search means the process of finding information on-line by the help of social resources, like, for example, asking to friends or unknown people at the Internet for answers [Morris et al 2010b].*

Definition 3 – Social Search by Evans & Chi: *Social Search is term used to describe search acts that make use of social interactions with other individuals. These interactions may be explicit or implicit, local or remote, synchronous or asynchronous [Evans & Chi 2010].*

Both definitions complement themselves. The definition 2 talks about the interactivity of social search while definition 3 discusses interaction types in social search. Following the concepts of social search projects like Aardvark [Horowitz & Kamvar 2010] and ChaCha [ChaCha 2012] were developed. The first one can search the users with the best knowledge to answer a question, while the second one offers a search service for answers made by specialists (its employees).

2.3. Memes

The term "meme" was introduced in 1967 by Richard Dawkins [Dawkins 2006]. It represents a cultural transmission unit that propagates itself in a society through the replication by imitation of things like ideas, clothes, styles, quotes, etc. For example, when a teacher reads something good, it's expected that he is going to spreads it among its students or colleagues. If what it has spread is something interesting eventually people will start to spread it too. Every time a meme reaches someone new, a popular meme (or fertile) can make that person a vehicle to propagate itself like a gene among individuals of the same specimen. Another analogy frequently used to explain what is a meme is comparing it to a virus. For example, a virus uses a host cells to propagate itself, like a meme uses the mind of its host to propagate itself.

While many memes can spread themselves many times, others can't do that. According to Dawkins [Dawkins 2006] it happens because of 3 memes characteristics (or dimensions): (1) fidelity; (2) fecundity; and (3) longevity. Fidelity is defined as the ability that a meme has to not change during the time even when it happens to be replicated many times. Fecundity is rate were a meme is replicated, the faster it is replicated, the more are the chances that it has to capture a large audience. Finally, longevity refers to the amount of time in which a meme can continue infecting minds.

3 Related Work

Despite the lack of projects directly related to our objective it's still possible to find in the literature some works using opinions in the development ranking functions. One of the first works in this field was published in 2006 [Mishne 2006]. It describes a ranking function associated with aspects (parameters) like sentiment analysis, spam detection, and link based authority estimation in order to rank blog posts. The function presented in this project is composed by a linear combination of those aspects. Moreover they have applied MAP (Mean Average Precision) and R-Precision metrics as the methodology to evaluate the weight of proposed parameters in the ranking function. At the end of the experiments they have concluded that the link based authority estimation value wasn't relevant improve to the ranking function while the other aspects worked well to improve results.

Another interesting project [Zhang 2009] proposes a ranking function to documents composed by two main values: opinion relevance and topic relevance. Both values are combined by a quadratic combination because the researchers believed that this type of combination is superior to linear interpolations. Based on that, they have developed a raking function and conducted experiments with TREC Blog06 and Blog07 corpus [Access to Web/Blog Research Collections 2012]. In those experiments they have applied as evaluation metrics MAP, R-Precision, and precision at top 10 (P@10) results in order to evaluate different variations of their ranking function. As a result they have concluded that the quadratic combination together with logarithm normalization was the best way to implement their ranking functions.

A third project [Huang and Croft 2009] presents a probabilistic ranking function for documents. The ranking function is based on the number of occurrence of opinion words. It's used together with others technics like query expansion based on a synonymous dictionary to build a score function for documents. The values were combined by a linear interpolation. Then they performed experiments in the TREC Blog06 and COEAE08 (a document dataset for the Chinese language compiled by the Chinese Academy of Sciences) databases to discover the weight of each component in their scoring function that maximizes the results. The main metric used is this project was MAP, however other metrics like R-Precision and P@10 where computed and presented in the paper.

The last project of our state of art review [Attardi and Simi 2006] presents a ranking function that uses as one of its parameters a relevance estimation function for opinions in documents. To achieve that, subjective words considered to be carrying an opinion bias were tagged in the documents. Then they build a system that are able receive as one of its inputs the subject that the user is searching for opinions. To interpret those queries their system finds for words carrying an opinion bias near to the words representing the subject. In order to evaluate the results they made experiments in the Blog06 using the P@5 (precision at top 5) as the main evaluation metric.

We believe that to develop a good ranking function exclusively for opinions (not for documents like blog posts) we can apply similar evaluation metric but use also concepts from research fields other than only information retrieval, like social search. How they are going to do that shall be described in the next section.

4. Model for Opinion Ranking

In order to create an abstract model to rank opinions we need to formally define the most important concept of our model, in particular what we consider an opinion.

Definition 4 – An Opinion *is a 7-tuple <author, date, site, sentence, subject, orientation, feature> where author is a string containing the name of the opinion's author, date is a date where the opinion was published, site is the URL (Universal Resource Locator) of the site where the opinion was published, sentence is a string with the sentence containing the opinion, subject is a string representing the opinion's subject, orientation is a string with the opinion's orientation ("positive", "negative" or "neutral"), feature is a string representing the feature which the opinion is about.*

Consider the following sentence about the electronic game Just Dance 2 that was published on 14 November 2010 by Alex St-Amour at vgchartz.com: *"The audio for this game is remarkably well done."* [St-Amour 2009]. In our model this opinion would be represented by the following 7-tuple: <"Alex St-Amour" , "14 November 2010", http://gamrreview.vgchartz.com/review/45671/just-dance-2/, "The audio for this game is remarkably well done.", "Just Dance 2 "positive", "audio">

We have been developing a model to rank opinions composed by a set of scores: (1) Information Retrieval Score; (2) Memetic Score; (3) Author's Experience; (4) Author's Image Score; (5) Author's Reputation Score; (5) Similarity Score; (6) Network Distance Score. Those concepts are described in the following sub-sections.

4.1. Parameters of our Model

The **Information Retrieval Score** is based on the basics IR concepts: term frequency and document frequency (tf-idf). Given a feature *f* and an opinion *o* it can be computed according to Equation 6.

$$IR\ Score(f,o) = \sum_{h \in Hyponymies(f)} tf(h, Sentence(o).idf(h))$$

Equation 6. IR Score.

Equation 6 sums the tf-idf score of every hyponym of the feature f returned by the function *Hyponimies*. We consider a hyponym as a non-empty set of words semantically linked to the feature. For example, in the context of electronic games, if f = "multiplayer" then *Hyponimies(f) = {"local multiplayer", "on-line multiplayer", "co-op multiplayer", "cooperative multiplayer", "competitive multiplayer"}.*

The **Memetic Score** is a social value computed by an estimation of the opinion's longevity. For example, if a positive opinion about a product feature has been said for many years, it could mean that positive opinions about that feature are relevant. Given an orientation *or*, Equation 7 shows how to estimate this concept. In Equation 7 *P(or|t)* represents conditional probability of a opinion with the orientation *or* given the time *t*. The concept of time in the equation can represent a month, a semester, a year, etc., it depends on the context where the model shall be applied. For example, if we take time a year and its initial value as 2000, Longevity will use *P(or|t=2000)*, *P(or|t=2001)*, *P(or|2002)*, etc. So, given an orientation, for each year the equation will compute and sum the variation of the probability that an opinion with that orientation occurs, compared to the previous year.

$$Longevity(or) = \sum_{t=0}^{n-1} -|P(or|t) - P(or|t+1)|$$

Equation 7. Longevity of an orientation.

The **Author's Experience** is a score reflecting the number of times that an author has expressed his opinion about a subject. Given an author *a*, a feature *f* and *Opinions* a function that returns a set of opinions containing only the opinions expressed by *a*, Equation 8 shows how to compute the author's experience.

$$Experience(a, f) = \sum_{op \in Opinons(a)} \begin{cases} 1, f = feature(op) \\ 0, f \neq feature(op) \end{cases}$$

Equation 8 – Author's Experience.

The **Author's Image Score** is a social value representing how a user sees an opinion's author as relevant or not. It can be estimated by user feedback while an opining search engine is running or by asking users about the authors before they start to use the system. Formally, given an author *a* and a user *u*, we will define the Image of *a* as a real number *v*, shown in Equation 9.

$$Image(a, u) = v, where\ v\ \in R\ and\ 0\ \ll v\ \ll 1$$

Equation 9. Author's Image.

The **Author's Reputation Score** is a social value defined by how users at the Internet evaluate an author relevant or not. It's basically the sum of all reputations of an author. Formally, given an author *a* and a set of all users *U*, Equation 10 shows how to compute the Image of *a*.

$$Reputation(a) = \frac{\sum_{u \in U} Image(a, u)}{|U|}$$

Equation 10. Author's reputation.

The **Similarity Score** is social concept defined by an estimation of preferences and interests that an author and a user have in common. For example, books, electronic games, hobbies, etc. Given an author *a*, a user *u* and *Interests*, a function that returns the set of preferences and interests that a person has interest, Equation 11 shows how to estimate the Similarity between *a* and *u*.

$$\text{Similarity}(a, u) = \frac{Interests(a) \cap Interests(u)}{Interests(u)}$$

Equation 11. Similarity score.

Finally**,** in a relationship graph what we call the **Network Distance Score** is the smallest number of nodes between the opinion's author and the user which made a query, it's is also a social metric. Formally, given a user *u*, an author *a* and *g*, a relationship graph where users and authors are the nodes, Equation 12 defines the distance among *a* and *u*.

$$\text{Distance}(u, a, g) = n, where\ n \in N\ and$$
$$n\ is\ the\ number\ of\ edges\ in\ a\ shortest\ path$$
$$connecting\ u\ and\ a\ in\ g$$

Equation 12. Network Distance Score.

4.2. An Opinion Raking Function

Formally it's possible to define our ranking function as follows:

Definition 5 – Opinion Ranking Function:

Inputs – An opinion; information retrieval score; memetic score; author's reputation score; author's image score; similarity between author and user/reader; network distance score; author's experience score.

Output – A score s (where s belongs to the set of real numbers) representing the opinion's score.

How those parameters will be combined is something that we expect to be different according to the domain where the model shall be applied. This is why we propose a way to discover a good combination of them through the use of an Artificial Intelligence technique called Genetic Algorithms [Russell and Norvig 2009]. This technique mimics the process of natural evolution by the simulation of concepts like: selection, inheritance, mutation, and crossover. We believe that it fits very well to solve our problem because genetic algorithms make it possible to test many combinations of our parameters in the heuristic way, without testing most of the possible permutations. So from an initial random set (initial population) of possible combinations for our parameters, applying selection, inheritance, mutation, and crossover it's possible to evolve them. This evolution should be made to achieve the best performance on the evaluation metrics usually applied to evaluate ranking functions (MAP, R-Precision, and precision at top 10).

Given the Longevity and Similarity functions (parameters of our model) and a, b, c, d (weights), in this first version of our model, we propose the following ways to combine our parameters:

- Linear combination: a.Longevity + b.Similarity
- Multiplication: a.Longevity.Similarity
- Logarithm combination: a.$\log_c$(Longevity) + b.$\log_d$(Similarity)
- Logarithm multiplication: a.$\log_c$(Longevity).$\log_d$(Similarity)

5. Model Discussion

In order to compare our approach to rank opinions with the ones applied by the other projects presented in the section 3, let us discuss how our function to rank opinions compares to them.

The first relevant difference is about the objectives: we want to rank opinions while the models that we could find in our literature review want to rank documents (sets of sentences that can eventually contain opinions). Thereby, let's take as example the Just Dance 2 review published at vgchartz.com [St-Amour 2010] once again. Our model will consider every single opinion as entity and then generates a score for each one of them. However, other models will handle the whole review as a single entity and generate a score for it based on the number of opinions on it. The objective of our model is to rank opinions, so all parameters that we consider are focused on the estimation of the opinions relevance. On the other hand, other projects while aiming to rank documents can usually apply other concepts like topic relevance. So the score computed by our model is exclusively linked to opinion's relevance, but the scores of other projects use opinion's relevance estimation as a parameter of a document relevance estimation function.

Regarding the concepts applied to rank an opinion, our model uses: tf-idf score, author's image, author's reputation, author's experience, orientation's longevity, similarity among user and author and finally network distance between user and author. While other models have score functions to evaluate an opinion based on to the number words carrying opinion bias. This contrast is a consequence of our model's objective, which leads its ranking function to explore many ways to rank an opinion. For example, given the sentence that we have analyzed before *"The audio for this game is remarkably well done."* [St-Amour 2010], for each sentence like these our model shall compute the score of all its parameters and then combine them. But the other models will just judge its relevance by the fact that it's an opinion or not since what they are ranking are documents.

Another important difference between our model and the previous models in the literature is the possibility to customize results. Social concepts [Morris at al 2010a] present in our project were developed to proper handle different kinds of users. When it comes to opinions it's expected that the same opinion can be relevant for a user A while it can be irrelevant for a user B. In the models that we could find it was impossible to provide different results for the same query. However in our model some of its concepts are inspired in Social Search (author's image, similarity and network distance) where the concepts defined allow customs results according to the user profile. For example, a sentence written by an author A can be on the top 10 best scored opinions for a user U1, because A has a good image for U1. However, for a user U2 the same sentence can't achieve the top 10 because U2 thinks that A doesn't have a good image.

There is a model that combines the scores of their secondary ranking functions in a final score by linear combination [Mishne 2006] while another uses a quadratic combination [Zhang & Ye 2009]. However our model only suggests possible ways to combine its concepts. We have made this decision because we believe that the best way to handle this issue is to combine changes according to the domain where the model shall be applied. Hence we also discuss how a way to find out a good combination of our parameters through Genetic Algorithms.

The different projects found in the literature developed their experiments using corpus composed mainly be blog posts, without the selection of a specific domain (for example, blogs about books). Since we believe that final result of a score function can vary according to its domain, we chose a specific type of opinions to evaluate: opinions about electronic games. Furthermore there are many other fields like books or music where opinions can the retrieved and ranked. However, when it comes to the evaluation metrics the related work have an important contribution to our work because they showed to us that it's possible to use information retrieval techniques in order to evaluate opinion mining systems, so we are able to apply traditional metrics like MAP to measure the results of our model.

6. Conclusion

Queries for opinions in traditional search engines are handled the same way as queries for facts. In order to fill this gap, developers have created a research area called Opinion Mining, whose goal is to handle the subset of queries designed to retrieve opinions. In this field, this report presented the current state of a model for scores of opinions by relevance, by combining several concepts that weren't present in models that we could find in our state of art review.

So far we have discussed mainly how to rank opinions. However, we have been applying our model to create a function in a specific domain: reviews for electronic games. But there are some tasks that we need to do in order to have a function ready to rank that kind of opinions. Among them it's possible to name: (1) to develop a hyponym thesaurus of electronic game features; (2) to train an opinion classifier; (3) to collect the relevance data from users; (4) to apply genetic algorithms for the best combination of the parameters; (5) and finally to make the proper adjustments in the model based on the experience acquired in the development of a ranking function for opinions about games. Currently, we have developed a crawler based on a collection of game reviews and we are also starting to develop an opinion classifier based on the work of [Li at al 2011], in parallel with the compilation of a hyponym thesaurus for games features..

References

"Access to Web/Blog Research Collections" (2012), http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html. March

Attardi, G., Simi , M. (2006) "Blog Mining through Opinionated Words". Fifteenth Text RE-trieval Conference.

ChaCha. (2012) "Questions & Answers | ChaCha", http://www.chacha.com/. March.

Dawkins, R. (2006) The Selfish Gene. Oxford University Press, USA, This 30th anniversary edition.

Evans, B. M., Chi, E. H. (2010) "An elaborated model of social search". Information Processing & Management Volume 46, Issue 6, November, Pages 656-678. Collaborative Information Seeking.

Google. (2012) "What you'll find with Search plus Your World", http://www.google.com/insidesearch/plus.html. March.

Horowitz, D., Kamvar, D. K. (2010) "The Anatomy of a Large-Scale Social Search Engine". 19th international conference on World Wide Web, pp. 431-440.

Huang, X., Croft, B. (2009) "A Unified Relevance Model for Opinion Retrieval". 18th ACM conference on Information and knowledge management.

Li, S., Wang, Z., Zhou, G., Lee, S. (2011) "Semi-Supervised Learning for Imbalanced Sentiment Classification". International Joint Conferences on Artificial Intelligence.

Liu, B. (2012) "Opinion Mining. Technical Report", http://www.cs.uic.edu/~liub/FBS/opinion-mining.pdf. March.

Liu, B. (2010) "Sentiment Analysis and Subjectivity". Workshop of Social Theory and Social Computing: Steps to Integration, March 22-2.

Manning, C. D., Raghavan, P., Schütze, H. (2008) Introduction to Information Retrieval. Cam-bridge University Press. Cambridge University Press.

Mishne, G. (2006) "Multiple Ranking Strategies for Opinion Retrieval in Blogs". The Fifteenth Text REtrieval Conference.

Morris, M. R., Teevan, J., Panovich, K. (2010a) "A Comparison of Information Seeking Using Search Engines and Social Networks". 4th International AAAI Conference on Weblogs and Social Media, pp. 291-294.

Morris, M. R., Teevan, J., Panovich, K. (2010b) "What Do People Ask Their Social Networks, and Why? A Survey Study of Status Message Q&A Behavior". 28th international conference on Human factors in computing systems.

Russell, S. Norvig, P. (2009) Artificial Intelligence: A Modern Approach. Prentice Hall. 3rd Edition.

St-Amour, A. (2010) "Just Dance 2 Review". http://gamrreview.vgchartz.com/review/45671/just-dance-2/. March.

Sherrets, D. (2008) "Google's Marissa Mayer: Social search is the future". VentureBeat, Jan 31, 2008. http://venturebeat.com/2008/01/31/googles-marissa-mayer-social-search-is-the-future/.

Zhang , M., Ye, X. (2009) "A Generation Model to Unify Topic Relevance". 31st annual international ACM SIGIR conference on Research and development in information retrieval.