Sentiment Analysis

Chapter · June 2013

DOI: 10.1007/978-1-4614-7202-5_4

Tiejian Luo
Chinese Academy of Sciences

Guandong Xu
University of Technology Sydney

Chapter 4
Sentiment Analysis

Sentiment analysis (also called opinion mining) refers to the application of natural language processing, computational linguistics, and text analytics to identify and classify subjective opinions in source materials (e.g., a document or a sentence). Generally speaking, sentiment analysis aims to determine the attitude of a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state (that is to say, the emotional state of the author when writing), or the intended emotional communication (that is to say, the emotional effect the author wishes to have on the reader).

Generally, sentiment analysis classifies text expressions in source materials into two types: (1) facts (objective): objective expressions about entities, events and their attributes, e.g., "I bought an iPhone yesterday"; and (2) opinions (subjective): subjective expressions of sentiments, attitudes, emotions, appraisals or feelings toward entities, events and their attributes, e.g., "I really love this new camera". It should be pointed out that not all subjective sentences contain opinions, e.g., "I want a phone with good voice quality"; and not all objective sentences contain no opinions, e.g., "The earphone broke in just two days!"

Therefore, it is important for sentiment analysis to identify and extract facts and opinions from source text materials. However, unfortunately, this is difficult to be achieved accurately. Let us consider the following example, which is a simple review on iPhone.

Example 1.1 (1) I bought an iPhone a few days ago. (2) It was such a nice phone. (3) The touch screen was really cool. (4) The voice quality was clear too. (5) Although the battery life was not long, that is ok for me. (6) However, my mother was mad with me as I did not tell her before I bought it. (7) She also thought the phone was too expensive, and wanted me to return it to the shop □.

In the above example, it is clear that the sentence (1) would be identified as a fact; and the sentences (2), (3) and (4) as opinions. However, for the sentences (5), (6) and (7), it may be difficult to automatically determine the opinion expressions on iPhone. Generally, opinions have targets (objects and their attributes) on which

T. Luo et al., *Trust-Based Collective View Prediction*,
DOI: 10.1007/978-1-4614-7202-5_4,

opinions are expressed. Below, we give some definitions to formally describe opinions [125].

Definition 1.1 (*Object*) an entity that can be a product, service, individual, organization, event, or topic, e.g., iPhone.

Definition 1.2 (*Attribute*) an object usually has two types of attributes: (1) components, e.g., battery, keypad/touch screen; and (2) properties, e.g., size, weight, color, voice quality.

Definition 1.3 (*Explicit and implicit attributes*) explicit attributes refer to those appearing in the opinion, e.g., "the battery life of this phone is too short"; and implicit attributes refer to those not appearing in the opinion, e.g., "this phone is too large" (on attribute size).

Definition 1.4 (*Opinion holder*) the person or organization that expresses the opinion.

Definition 1.5 (*Opinion orientation*) i.e., polarity, e.g., positive, negative or neutral.

Definition 1.6 (*Opinion strength*) level/scale/intensity of opinion indicating how strong it is, e.g., contented, happy, joyous and ecstatic, whose strength are incremental.

Definition 1.7 (*Opinion*) a person or organization that expresses a positive or negative sentiment on a particular attribute of an object at a certain time, thus, which can be represented as a quintuple: $<$ object, attribute, orientation, opinion holder, time $>$.

However, in Definition 1.7 about the quintuple on opinions, it should be pointed out that (1) some information may be implied due to pronouns, context, or language conventions; (2) some information available from document attributes; and (3) in practice, not all five elements are needed. Besides, options can be classified into: (1) direct opinion, sentiment expressions on one or more attributes of an object, e.g., "the voice quality of this phone is fantastic"; and (2) comparative opinion, relations expressing similarities or differences between two or more objects based on some of the shared attributes of the objects, e.g., "the voice quality of camera x is better than that of camera y". However, from the viewpoints of opinion expressions, they also can be can be classified into: (1) explicit opinion, an opinion on an attribute explicitly expressed in a subjective sentence, e.g., "the voice quality of this phone is amazing"; and (2) implicit opinion, an opinion on an attribute implied in an objective sentence, e.g., "the headset broke in two days".

A main goal in sentiment analysis is to identify and classify options in source text expressions, i.e., to identify whether given document (e.g., product reviews, blogs, forum posts) or a given sentence expresses opinions and whether the opinions are positive, negative, or neutral. More generally, the basic tasks of sentiment analysis mainly includes: (1) sentiment identification, i.e., subjectivity identification, which aims to identify whether a piece of text expresses opinions;

and (2) sentiment orientation classification, which aims to determine the orientation of an opinionated text. In the following sections, we briefly review some methods on the above two tasks.

4.1 Sentiment Identification

Sentiment identification (also called subjectivity identification) refers to identify whether a piece of text (e.g., a document, a sentence) expresses opinions, generally, which is based on the following two basic assumptions: (1) the given text is opinionated on a single object; and (2) the opinions are from a single opinion holder.

In sentiment identification, the main task is to identify opinion words, which is very important. Opinion words are dominating indicators of sentiments, especially adjectives, adverbs, and verbs, e.g., "I absolutely love this camera. It is amazing!" Opinion words are also known as polarity words, sentiment words, opinion lexicon, or opinion-bearing words, which generally can be partitioned into two types: (1) positive words, e.g., wonderful, elegant, amazing; and (2) negative words, e.g., horrible, disgusting, poor. In order to identify opinion words from a given piece of text, we need to in advance generate a set of opinion words. If we have known a rich set of opinion words, the sentiment identification process is easy to be achieved, which only need to search out each opinion word from source text expressions.

Aiming at generating opinion words, there are three types of methods: (1) manual generation method, i.e., collecting opinion words manually, obviously, which is effective (i.e., accurate) but expensive; (2) dictionary-based generation method, which uses a seed list and grow the list, e.g., SentiWordNet [126]; and (3) corpus-based generation method, i.e., relying on syntactic or co-occurrence patterns in large text corpora. Below, we briefly review the latter two types of methods (i.e., dictionary-based and corpus-based) on how to generate opinion words.

4.1.1 Dictionary-Based Opinion Words Generation

In this method, a list of seed opinion words is needed to prepare in advance; and then a dictionary is used to help to grow the list to generate more opinion words. More specifically, the method of dictionary-based opinion words generation can be described as follows:

1. A small seed set of opinion words with known orientations is collected manually, e.g., {"glad"}, a set of a positive opinion word "glad".

2. An online dictionary (e.g., WordNet [127]) is then searched for their synonyms and antonyms to grow the seed set of opinion words, for example, after adding the synonyms of "glad", we obtain a set of more positive opinion words: {"glad", "happy", "joyful", "delighted"}; and after adding antonyms of "glad", we obtain a new set of negative opinion words: {"sad", "unhappy", "sorry", "heart-broken"}.
3. The above two steps would be repeated until no more new opinion words can be found from the online reference dictionary.
4. Finally, manual inspection may be done for correction.

In addition, in the above process, some additional information (e.g., glosses) from WordNet can be used. Overall, the process of dictionary-based opinion words generation is simple and easy to be understood. However, this method cannot identify context-dependent opinion words, i.e., the words whose opinion orientations heavily rely on the context. Let us consider the following two examples.

Example 2.1 Given an opinion word "small", and two sentences "The LCD screen is too small" and "The camera is very small and easy to carry", we can find that in the first sentence, the opinion word "small" is negative, and in the second sentence, it is positive, i.e., the opinion orientation of the word "small" is context-dependent □.

Example 2.2 Given an opinion word "long", and two sentences "It takes a long time to focus" and "The battery life is long", we can find a situation similar to Example 2.1 is encountered □.

4.1.2 Corpus-Based Opinion Words Generation

Different to the dictionary-based generation, corpus-based opinion words generation depend on syntactic or co-occurrence patterns in large text corpora [128]. One of the main advantages of corpus-based opinion words generation over dictionary-based generation is that it can obtain domain dependent orientations and/or context dependent ones [125]. Below, we give an example of identifying the sentiment orientations of adjectives in context.

1. Start with a list of seed opinion adjective words.
2. Based on a chosen corpus, use linguistic constraints on connectives to identify additional adjective opinion words and their orientations, where the linguistic constraints include: (1) sentiment consistency, which is based on the observation that conjoined adjectives usually have the same orientations, e.g., given a sentence "This car is beautiful and spacious", if the word "beautiful" is positive, then "spacious" is positive too. (2) some rules can be designed for different connectives, e.g., AND, OR, BUT, EITHER-OR, NEITHER-NOR.
3. Use log-linear model to determine if two conjoined adjectives are of the same or different orientations.

4. Use clustering to produce two sets of opinion words, i.e., positive words and negative words.

Using the above process, we can determine domain opinion words. However, finding domain opinion words is not sufficient. One opinion word may indicate different opinions in the same domain, e.g., "The battery life is long" versus "It takes a long time to focus". To overcome this problem, a basic idea is to identify all opinion words related to a specific object attribute, whose process can be briefly described as follows (see [129] for more detail): (1) create pairs of object attribute and opinion word, i.e., < object attribute, opinion word > ; and (2) then, determine opinion words and their orientations together with the object attributes. Based on the above process, the context dependency of opinion words can be handled.

4.2 Sentiment Orientation Classification

Sentiment orientation classification refers to determine the opinion orientation of an opinionated text, i.e., based on the opinion words identified from a given piece of text by sentiment identification, to determine whether the opinion orientation in the given text are positive, negative or neutral.

Based on the assumption that the given text is opinionated on a single object, that is to say, all the opinion words identified from the given text act on a single object, a straightforward way of classifying sentiment orientation is to count positive and negative opinion words, e.g., in Example 1.1, the number of positive opinion words on the object "iPhone" is greater than the number of negative words, so it can be simply considered that the review expresses positive sentiment orientation. Besides, machine learning approaches also can be used for sentiment orientation classification. Below, we briefly review two types of methods on sentiment orientation classification (i.e., counting opinion words and supervised machine learning).

4.2.1 Counting Opinion Words

Counting opinion words is a simple method on sentiment orientation classification, which is based on the predefined opinion words generated by sentiment identification. In this method, first, we need to assign orientation score (+1, −1) to all opinion words: (1) positive opinion words (+1), e.g., great, amazing, love; and (2) negative opinion words (−1), e.g., horrible, hate. In this process, we also can use strength value between [0, 1], based on the opinion strength of opinion words. Last, the opinion orientation score of a given piece of text is simply considered to be equal to the sum of orientation scores of all opinion words found. Let us consider the review on iPhone in Example 1.1. If we assign (+1) to each positive opinion word (i.e., “nice”, “cool”, “clear” and “ok”) and (-1) to the negative opinion word(i.e., “expensive”), then the review has an opinion orientation score of $4-1 = 3$, and thus it has positive opinion orientation.

However, the above way by simply counting opinion words is obviously not accurate enough, for example, given a sentence “There is not one thing I hate about this product”, it would be assigned with a negative opinion orientation score, but in fact it is positive. To overcome the problem caused by negation words, we can create some basic opinion rules to improve the accuracy of simply counting opinion words, for example, two simple rules can be manually created as follows: both “not … negative” and “never … negative” represent positive opinions. See [125] for more detail on rule-based opinion words counting methods.

4.2.2 Supervised Learning Approaches

Using counting opinion words, generally, only a limited number of opinion words can be found, and only a limited number of patterns can be created. Can we automate the task with limited manual work (e.g., find opinion words and their orientations automatically)? To solve this, supervised learning approaches are advocated to apply into sentiment orientation classification. Here, the basic idea is to leverage supervised learning techniques to find patterns in known examples and apply them to new documents, so as to classify the sentiment orientation of new documents automatically. Thus, the goal of supervised learning is to train and obtain an opinion classifier which contains some target opinion classes, e.g., positive versus negative. Most existing supervised learning approaches can be used to achieve the goal (i.e., to obtain the opinion classifier). Now, popular supervised learning methods mainly include:

1. Naïve Bayes (NB): A simple probabilistic classifier based on applying Bayes’ theorem with strong (naive) independence assumptions.
2. Maximum Entropy (ME): A probabilistic model that estimates the conditional distribution of the class label.
3. Support Vector Machines (SVM) [130]: A representation of the examples as points in space in which support vectors are computed to provide a best division of points/examples into categories.
4. Logistic Regression Model (LR) [131]: A LR model predicts the classes from a set of variables that may be continuous, discrete or a mixture.

However, a classifier obtained by supervised learning approaches often lead to a problem of domain dependency, i.e., the classifier trained using opinionated documents from domain A often performs poorly when tested on documents from domain B. Generally, this problem is caused due to the following two reasons [132]: (1) words used in different domains can be substantially different, e.g., Cars versus movies, and Cameras versus Strollers; and (2) some words mean opposite in two domains, e.g., “unpredictable” may be negative in a car review, but positive in a movie review, and “cheap” may be positive in a travel/lodging review, but negative in a toys review. This problem can be solved by the approach in [133], which is briefly described as follows :

1. Use labeled data from one domain and unlabeled data from both source the target domain and general opinion words as features.
2. Choose a set of pivot features which occur frequently in both domains.
3. Model correlations between the pivot features and all other features by training linear pivot predictors to predict occurrences of each pivot in the unlabeled data from both domains.

4.3 Case Study: Sentimental Analysis in Recommender Systems

4.3.1 Introduction

Recently, social tagging systems originated with sites such as Del.icio.us[1] and Flickr[2] have become a popular trend in Web 2.0 environment. Social tagging systems are designed for the storage, organization, retrieval and share of personal resources such as links, videos and photos on the web. By tagging, users annotate and index the interested resources freely and subjectively, based on their senses of interests. As such, these user generated tags can be seen as a new metadata carrying on the user preference and resource relatedness in social collaborative systems.

Nowadays people are inundated by information loads and choices. Recommender systems are proposed as a solution to this problem by providing users with the interested and needed information. Various kinds of recommender systems [134, 135] are developed for better user experience, in which the personalized recommendation takes an important aspect within the whole recommendation approaches. Traditional recommender systems focus on the explicit rating data of users, e.g., movie ratings, to gain the user-interest profile and make predictions for new items. Different from the rating data, social tagging data does not contain user explicit preference information on resources, instead, reflecting the implicit perceptions on certain resources by annotating their opinions or perceptions. Therefore, making use of tagging data as an additional feature will undoubtedly facilitate the capture of user preference and resource relatedness for better personalized recommendation. Recently social tag recommender system is emerging as an active research topic in the domain of recommender systems.

1 http://del.icio.us

2 http://flickr.com

As discussed above, the aim of recommendation is to improve the user experience and satisfaction when they are browsing the web and interacting with various systems for specific information needs. Thus in recommendation practice we sometimes need to recommend the items or resources that gained positive feedbacks or reflected preferable comments in addition to the intuitive similarities on some aspects such as categorical and contextual information. The tagging data that annotated by users on resources, of course, is able to provide such kind of polarity attributes in social annotation systems. Intuitively, we aim to incorporate sentiment analysis into social tag recommender systems to improve current recommendation approaches.

Some related studies have been conducted to reveal the subjective inherence of tagging in terms of sentiment (or opinion) expression. For example, Golder and Huberman [136] found evidences of sentiment expression in the self-tagging site Del.icio.us. Zollers [137] discovered the sentiment expression in social tagging systems like Amazon.com and Last.fm (shown in Tables 4.1 and 4.2).

All of the sentimental tags found in Last.fm and Amazon.com along with other co-occurred noun tags which described the resource that was being tagged indicate the polarized opinions of users on these resources. Additionally, it is possible to see that the expressed opinions also work as indicators in recommendations either for or against a resource.

Traditionally, sentiment analysis refers to classifying text documents, such as user reviews, newsgroup messages and blogs, based on the polarity of the opinions they express [47]. Specifically, sentiment analysis is concerned with the automatic identification, extraction, and classification of opinions in texts. It can be used to develop applications that assist decision makers and information analysts in tracking user opinions about topics that they are interested in. Examples of sentiment analysis include the classification of a movie review as "thumbs up" or "thumbs down" [132].

Motivated by the intuition of sentiment analysis, in this research we aim to propose a new recommendation approach by incorporating sentiment factor of tagging into social tag recommender systems. On top of conventional social tag recommender systems, the tag sentiment information presented in the form of

Table 4.1 Sample opinion tags found on last.fm

Opinion tag	Tag data
Awesome	7, 034 people used this tag 36, 363 times
Beautiful	6, 865 people used this tag 38, 510 times
Great	2, 028 people used this tag 7, 756 times
Crap	1, 520 people used this tag 4, 950 times

Table 4.2 Sample opinion tags found on Amazon.com

Opinion Tag	Tag data
Awesome	691 people used this tag on 855 items
Beautiful	299 people used this tag on 356 items
Cool	404 people used this tag on 574 items

subjective polarity of the user toward the annotated resources, work like an additional information filtering to recommend more preferable and positive resources to the user. Therefore, the sentimental tags can be used to improve the performance of recommendation. To our best knowledge, there is very little work addressing the sentiment analysis enhancement for improved tag-based recommendations in previous studies.

4.3.2 Related Work

4.3.2.1 Tag-Based Personalized Recommendation

Durao and Dolog [138] developed a multi-factorial tag-based recommender system, which took various lexical and social factors of tags into the similarity calculation. Using tags as a means to express which features of an item user particularly like or dislike, Gedikli and Jannach [139] proposed a simple recommendation method that can take item-specific tag ratings into account when generating rating predictions. Zhao et al. [140] proposed a collaborative filtering approach Tag-based Collaborative Filtering (TBCF) based on the semantic distance among tags assigned by different users to improve the effectiveness of neighbor selection. Shepitsen et al. [141] proposed a personalized recommendation system by using hierarchical clustering. In this approach, instead of using the pure tag vector expressions, a processing on tag clustering was performed to find out the tag aggregates for personalized recommendation. Xu et al. [142] proposed a Semantic Enhancement Recommendation strategy (SemRec), based on both structural information and semantic information through a unified fusion model, which combines the clustering and hidden topic model. Extensive experiments conducted on two real datasets demonstrate the effectiveness of their approaches. Leung et al. [143] described a rating inference approach to incorporating textual user reviews into collaborative filtering (CF) algorithms. The main ideas of their approach is to elicit user preferences expressed in textual reviews, a problem known as sentiment analysis, and map such preferences onto some rating scales that can be understood by existing CF algorithms.

4.3.2.2 Sentiment Analysis

In the literature, sentiment analysis goes under various names, such as opinion mining, sentiment mining. Its related work may come from both computer science and linguistics. The task of sentiment analysis can be roughly divided into three sub-categories: determining subjectivity [144], determining orientation, and determining strength of orientation [145], and most of the studies focus on investigating the sentiment orientation of words, phrases, and documents. Turney [132] used point-wise mutual information (PMI) to calculate an average semantic orientation score of extracted phrases for determining the document's polarity. Amps et al. [146] tried to evaluate the semantic distance from a word to good/bad with WordNet. Pang et al. [130] employed three machine learning approaches to annotate the polarity of IMDB movie reviews. Jindal and Liu [48] built a framework to compare consumer opinions of competing products using multiple feature dimensions. After deducting supervised rules from product reviews, the strength and weakness of the product were visualized with an Opinion Observer. Scaffidi et al. [145] presented a search system called Red Opal that examined prior customer reviews, identified product features, and then scored each product on every feature. Zhuang et al. [147] presented a supervised approach for extracting feature-opinion pairs. Their method learned the opinion and a combination of dependency and part-of-speech paths connecting such pairs from an annotated dataset.

4.3.3 Preliminaries

4.3.3.1 Social Tagging System Model

In this study, our work is to deal with tagging data. A typical social tagging system has three types of entities, users, tags and resources which are interrelated with one another. Social tagging data can be viewed as a set of triples [148]. Each triple $\left(U_i, T_j, R_k\right)$ represents an observation of a user U_i annotating a tag T_j on a resource R_k. A social tagging system can be described as a four-tuple—there exist a set of users, U; a set of tags, T;a set of resources, R; and a set of annotations, A. We denote the data in the social tagging system as D and define it as: $D = \langle U, T, R, A \rangle$. The annotations, A, are represented as a set of triples containing a user, tag and resource defined as: $A \subseteq \left\langle U_i, R_j, R_k \right\rangle : U_i \in U, T_j \in T, R_k \in R$.

4.3.3.2 Standard Tag-Based Recommendation

The standard tag-based recommendation is principally similar to a process of traditional information retrieval but with an additional input of the user tagging preference for personalization (or called personalized recommendation). The procedure consists of two steps of search and personalization. The search step produces a list of candidate resources based on the similarity computation between the query tag issued by a user and all resources in terms of term frequency—inverse document frequency (tf-idf).

The second step utilizes the tagging preference of users to make the personalization. Under the vector space model, each user, u, is modeled as a vector (also called user profile) over the set of tags, where $w(T_i)$, in each dimension corresponds to the relationship of a tag t_i with this user, U_i $U_i = \left\langle w_i(T_1), w_i(T_2), \cdots, w_i\left(T_{|T|}\right) \right\rangle$.

Likewise each resource, R_j, can be modeled as a vector (i.e., resource profile) over the same set of tags, $R_j = \left\langle v_j(T_1), v_j(T_2), \cdots, v_j\left(T_{|T|}\right)\right\rangle$. After that, the similarity computation, e.g., cosine measure, of the target user profile u and the candidate resource profiles r selected by the first step, is performed, $sim\left(U_i, R_j\right)$, to further generate the personalized resources based on various recommendation strategies. The distinction of the tag-based recommendation from the standard information search is that here the recommendation is derived upon, not only the query itself, but also the user tagging preference (i.e., personalization).

4.3.4 Sentiment Enhanced Approach for Tag-Based Recommendation

4.3.4.1 Sentiment Enhanced Approach

In order to generate personalized recommendations, Durao et al. proposed a framework for the calculation of similarity between resources based on tags. Their method combines the basic cosine similarity calculus with other factors, such as tag popularity, tag representativeness and an affinity user-tag for the purpose of reordering the original raking in recommendation and generates personalized ones. Based on their work and incorporating a sentiment factor, we present an approach to calculating the similarity of resources as follows:

$$Sim(R_A, R_B) = [(D_A + D_B) * cos\_sim(R_A, R_B)] * sentiment(UR_A,\ \mathrm{R}_B) \tag{4.1}$$

where, R_A and R_B are the resources in a social tag system. If a user UR labels the resource R_A by tagging, and resource R_B is the other resource in the system, Eq. (4.1) calculates the similarity score between R_A and R_B. We then select the Top-N score resources for the user UR who labels the resource A as the personalized recommendation.

In Eq. (4.1), D_A and D_B is the score of resource R_A and R_B, D_A is defined as follows:

$$DA = \sum_{i=1}^{n} (weight(Ti) * representativness(Ti)) \tag{4.2}$$

where n is the total number of tags labeled on R_A. The $weight(T_i)$ factor is the popularity of the T_i in the social tagging system, which is calculated as a count of occurrences of one tag per total of resources available. We rely on the fact that the most popular tags are like anchors to the most confident resources. As a consequence, it decreases the chance of dissatisfaction by the receivers of the recommendations.

The $representativeness(T_i)$ factor measures how much a tag can represent a document it belongs. It is believed that those tags which appear most in the document can better represent it. The tag representativeness is measured by the term frequency, a broad metric also used by the information retrieval community.

The $\cos\_sim(R_A, R_B)$ factor in Eq. (4.1) is a cosine similarity between resource R_A and R_B from the classical text mining and information retrieval domain, where, two resources are thought of as two vectors in the m-dimensional tag-space under the vector space model. The similarity between them is measured by computing the cosine function of the angle between these two vectors.

The $sentiment(UR_A, R_B)$ in Eq. (4.1) is a sentiment enhanced factor proposed in our approach, which is defined as follows:

$$sentiment(URA, R_B) = \sum_{i=1}^{n} sentiment\_score(Ui, R_B) \tag{4.3}$$

where, $UR_A = \{U_1, U_2, \ldots, U_N\}$ is a set of Top-N nearest neighboring users of user UR who labels the resource R_A. The similarity between two users is calculated using cosine similarity based on tag-vector space model same as $\cos_sim(R_A, R_B)$factor.

$sentiment\_score(U_i, R_B)$ in Eq. (4.3) is the total sentiment score of user U_i towards resource R_B as follows:

$$sentiment\_score(Ui, R_B) = \sum_{i=1}^{k} sentiwordnet(Ti) \tag{4.4}$$

where, T_i is a tag labeled on resource R_B by user $U_i \cdot sentiwordnet(T_i)$ is a polarity score of T_i calculated based on the Senti WordNet, a public available resource for sentiment analysis evolved from Wordnet, which is a wildly used lexical database of English. In SentiWordNet, each WordNetsynset is assigned with a triplet of numerical scores representing how Positive, Negative and Objective a synset is. SentiWordNetis proved as a useful tool for sentiment mining applications, because of its wide coverage (all WordNetsynsets are assigned according to each of the three labels Objective, Positive, Negative) and because of its fine granularity, obtained by qualifying the labels by means of numerical scores.

In our approach, $sentiwordnet(Tag_i)$ in Eq. (4.4) is defined as follows:

$$sentiwordnet(T_i) = \max_{positive}(T_i) - \max_{negative}(T_i) \tag{4.5}$$

where, $\max_{positive}(T_i)$ and $\max_{negative}(T_i)$ are the maximum positive and negative polarity score of synsets of T_i Normally, a tag have a set of different synsets in Wordnet, therefore, T_i have multiple different scores of positive and negative polarity score for each synset. In order to emphasize the effect of the sentiment enhanced factor of a tag labeled on the resource, we choose the maximum positive and negative polarity score of synsets of T_i as calculating parameters in Eq. (4.4).

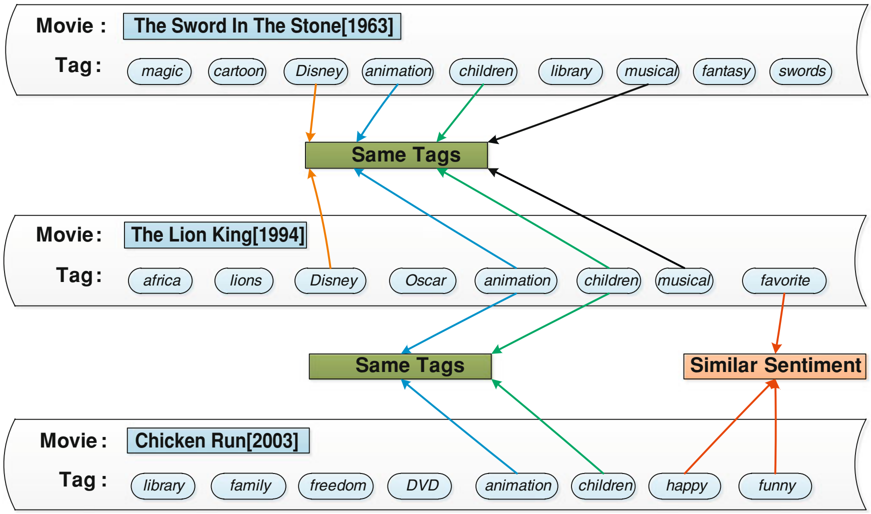

Fig. 4.1 Example of sentiment enhanced tag recommendation

4.3.4.2 Running Example

Now, let us look at a running example which explains our approach and ideas behind it. According to Fig. 4.1, three movies "The Sword In The Stone [1963]", "Chicken Run [2003]", "The Lion King [1993]" and their tags taken from MovieLens are shown. In this example, The "The Lion King" and "Sword In The Stone" have four same tags. "The Lion King" and "Chicken Run" only have two same tags, but they own three sentiment tags with same positive polarity. The polarity scores of three tag calculated from Eq. (4.5) are shown in Table 4.3. Here, one synset of the tag "*funny*" is "experiencing odd bodily sensations", therefore, the $\max_{negative}(funny)$ value of the word "funny" is larger than $\max_{positive}(funny)$ value.

If the sentiment enhanced factor is not considered, *Similarity* [The Lion King (1993), The Sword In The Stone (1963)] is larger than *Similarity* [The Lion King (1993), Chicken Run (2003)], otherwise, *Similarity* [The Lion King (1993), Chicken Run (2003)] is larger than *Similarity* [The Lion King (1993), The Sword In The Stone (1963)]. Therefore, Chicken Run (2003) will be recommended to the user who labels The Lion King (1993) in our approach, rather than The Sword In The Stone (1963).

Table 4.3 Sentiment polarity scores of tags

Tag	Positive score	Negative score
Favorite	0.375	0.125
Happy	0.875	0
Funny	0.5	0.625

4.3.5 Experimental Evaluation

To evaluate our approach, we conducted evaluation experiment son the popular MovieLens data set using a common experimental procedure and well known precision metrics. We performed the experiments using an Intel Core i3 CPU (2.13 GHz) workstation with a 2G memory, running windows 7. All the algorithms were written in Java. The results of this evaluation are described in this section.

4.3.5.1 Experimental Data Set

Our experimental data set is “MovieLens10M Ratings 100kTags” dataset, which is provided by GroupLens. It is a standard movie rating dataset used to study recommendation engines, tagging systems, and user interfaces. Data set consists of three files: ratings, movies and tags. The ratings file contains a list of user ratings on a 5-star scale with half-star increments. The movies file contains information about each movie such as the title and the genre, which are, however, not used in our method. The tags file contains the information about which tags have been assigned by the users to the movies. A tag assignment is a triple consisting of one user, one movie an done tag. Data set contains 10, 000, 054 ratings and 95, 580 tags applied to 10, 681 movies by 71, 567 users of the online movie recommender service MovieLens.

4.3.5.2 Data Set Preprocessing

Tag quality is one of the major issues when developing and evaluating approaches that operate on the basis of user-contributed tags. Reference [149] revealed that only 21 % of the tags in the MovieLens system had adequate quality to be used to evaluate approach.

A data pruning measures is applied to improve the quality of the existing tag information in our experiment. We defined the following two requirements for movies, tags and users to be taken into account in our evaluation. First, we only consider movies that have at least 15 tags assigned. Second, only those users are considered that they rate at least 10 movies to avoid problems with memory limitations. The resulting dataset used in our experiments finally includes 3,390 movies, 1,151 users and 2,645 tags shown in Table 4.4.

Table 4.4 Statistics of experiment datasets

Number of users	1,151
Number of movies	3,390
Number of tags	2,645
Minimum movies per user	10
Minimum tags per movies	15

4.3.5.3 Evaluation Methodology and Results

The goal of our analysis is to determine whether the sentiment enhanced recommending approach is effective. We compared the following algorithms:

- TBR-CS(tag-based recommender with cosine similarity): it uses the similarity metric $Sim(R_A, R_B) = [(D_A + D_B) * cos\_similarity(R_A, R_B)]$ in Eq. (4.1).
- TBR-AF(tag-based recommender with affinity factor): it uses the similarity metric $Sim(R_A, R_B) = [(D_A + D_B) * cos\_similarity(R_A, R_B)] * AF\left(U_i, T_j\right)$ as Durao et al. proposed.
- TBR-SF(tag-based recommender with sentiment factor): it uses the similarity metric
- $Sim(R_A, R_B) = [(D_A + D_B) * cos\_sim(R_A, R_B)] * sentiment(UR_A, R_B)$ as Eq. (4.1) proposed.

We utilized the standard metric from the area of information retrieval to evaluate our approaches. For each dataset, we randomly and judiciously divided the whole dataset into two parts by 80 (Training set) and 20 % (Test set). Here we use precision as evaluation metrics. In precision evaluation, for each given user from the test set, we determine the Top-N resources as recommendations based on the generated sentiment enhanced similarity scores. Then we count the total number of resources which are simultaneously occurred in the recommended resource list and real test set for each user and calculate the ratio of this number to the recommendation size as precision. Eventually we average the precision values over the total test set to obtain the final evaluation result.

Here, we report the experimental results of improved recommendation performance in comparison to two baseline approaches, i.e., using pure cosine similarity metric(TBR-CS) and affinity enhanced metric(TBR-AF). With the parameter $N = 50$, an example with the top 5 movies recommended to the user identified by User-ID 146 who annotated the movie "Star Wars Episode I" by three algorithms is shown in Table 4.5. From the example, we can see that the recommendation made by approach TBR-AF looks more reasonable, e.g., "*Star Wars Episode IV*" is ranked at the top.

The overall precision results of three recommendation approaches are shown in Table 4.6. From Table 4.6, we can see that our (TBR-SF) approach proposed in this study consistently outperforms both pure cosine similarity based method as

Table 4.5 Top 5 recommendation results to User-ID 146

Approach	REC	TBR-CS	TBR-AF	TBR-SF
User-ID 146/	@top-1	X-men	Stargate	Star wars episode IV
Star wars episode I	@top-2	Terminator	Star wars episode IV	X-men
	@top-3	Stargate	X-men	Terminator
	@top-4	Alien	Terminator	Stargate
	@top-5	Star wars episode IV	Alien	Alien

Table 4.6 Overall precision of three approaches

Approach	Precision (%)
TBR-CS	25.12
TBR-AF	27.31
TBR-SF	29.70

well as the affinity enhanced method in terms of precision, and the improvements are around 18.2 and 8.7 %, respectively. As a result, we conclude the proposed sentiment enhancement approach is able to achieve the better recommendation outcomes.