© 2017 IJSRSET | Volume 3 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

A Semantic Metadata Enrichment Software Ecosystem based on Sentiment and Emotion Metadata Enrichments

Ronald Brisebois[1], Alain Abran[1], Apollinaire Nadembega*[2], Philippe N'techobo[3]

[1]École de technologie supérieure, University of Quebec, Montreal, Quebec, Canada

*[2]Network Research Lab., University of Montreal, Montreal, Quebec, Canada

[3]École Polytechnique de Montréal, Montreal, Quebec, Canada

ABSTRACT

Information retrieval and analysis is frequently used to extract meaningful knowledge from the unstructured web and long texts. As existing computer search engines struggle to understand the meaning of natural language, semantically sentiment and emotion enriched metadata may improve search engine capabilities and user finding. A semantic metadata enrichment software ecosystem (SMESE) has been proposed in our previous research. This paper presents an enhanced version of this ecosystem with a sentiment and emotion metadata enrichments algorithm. This paper proposes a model and an algorithm enhancing search engines finding contents according to the user interests, through text analysis approaches for sentiment and emotion analysis. It presents the design, implementation and evaluation of an engine harvesting and enriching metadata related to sentiment and emotion analysis. It includes the SSEA (Semantic Sentiment and Emotion Analysis) semantic model and algorithm that discover and enrich sentiment and emotion metadata hidden within the text or linked to multimedia structure. The performance of sentiment and emotion analysis enrichments is evaluated using a number of prototype simulations by comparing them to existing enriched metadata techniques. The results show that the algorithm SSEA enable greater understanding and finding of document or contents associated with sentiment and emotion enriched metadata.

Keywords: Emotion Analysis, Natural Language Processing, Semantic Metadata Enrichment, Sentiment Analysis, Text And Data Mining

I. INTRODUCTION

Semantic information retrieval (SIR) is the science of searching semantically for information within databases, documents, texts, multimedia files, catalogues and the web. The human brain has an inherent ability to detect sentiment and emotion in written or spoken language. However, the internet, social media and repositories have expanded the number of sources, volume of information and number of relationships so fast that it has become difficult to process all this information [1].
Finding bibliographic references or semantic relationships in texts makes it possible to localize specific text segments using ontologies to enrich a set of semantic metadata related to sentiment or emotion. This paper presents an enhanced SMESE model and prototype [2] using metadata from linked open data, structured data, metadata initiatives, concordance rules and authorities metadata.

The current methodology proposed by SIR researchers for text analysis within the context of entity metadata enrichment (EME) reduces each document in the corpus to a vector of real numbers where each vector represents ratios of counts. Several EME approaches have been proposed, most of them making use of term frequency–inverse document frequency (tf-idf) [3, 4]. In the tf-idf scheme, a basic vocabulary of "words" or "terms" is chosen, then for each document in the corpus, a frequency count is calculated from the number of occurrences of each word [3, 4]. After suitable normalization, the frequency count is compared to an inverse document frequency count (e.g the inverse of the number of documents in the entire corpus where a

given word occurs — generally on a log scale, and again suitably normalized). The end result is a term-by-document matrix X whose columns contain the tf-idf values for each of the documents in the corpus. Thus the tf-idf scheme reduces documents of arbitrary length to fixed-length lists of numbers. For non-textual content, tools are available to extract the text from multimedia entities. For example, Bougiatiotis and Giannakopoulos [5] propose an approach that extracts topical representations of movies based on mining of subtitles. This paper focuses on contributions to mainly one EME research fields: sentiment analysis (SA) including emotion analysis.

The main objective of SA is to establish the attitude of a given person with regard to sentences, paragraphs, chapters or documents [1, 4, 6-12]. Indeed, many websites offer reviews of items like books, cars, mobiles, movies etc., where products are described in some detail and evaluated as good/bad, preferred/not preferred; unfortunately, these evaluations are insufficient for users in order to help them to make decision. In addition, with the rapid spread of social media, it has become necessary to categorize these reviews in an automated way [4]. For this automatic classification, there are different methods to perform SA, such as keyword spotting, lexical affinity and statistical methods. However, the most commonly applied techniques to address the SA problem belong either to the category of text classification supervised machine learning, which uses methods like naïve Bayes, maximum entropy or support vector machine (SVM), or to the category of text classification unsupervised machine learning (UML). Also, fuzzy sets appear to be well-equipped to model sentiment-related problems given their mathematical properties and ability to deal with vagueness and uncertainty — characteristics that are present in natural languages processing.

Thus, a combination of techniques may be successful in addressing SA challenges by exploiting the best of each technique. In addition, the semantic web may be a good solution for searching relevant information from a huge repository of unstructured web data [6].

According to [7], the SA process typically consists of a series of steps:

1. Corpus or data acquisition
2. Text preprocessing
3. Opinion mining core process
4. Aggregation and summarization of results
5. Visualization

One current limitation in the area of SA research is its focus on sentiment classification while ignoring the detection of emotions. For example, document emotion analysis may help to determine an emotional barometer and give the reader a clear indication of excitement, fear, anxiety, irritability, depression, anger and other such emotions. For this reason, our research focuses on sentiment and emotion analysis (SEA) instead of SA.

A number of algorithms are used to perform text mining, including: latent Dirichlet allocation (LDA) [13], tf-idf [3, 4], latent semantic analysis (LSA) [14], formal concept analysis (FCA) [15], latent tree model (LTM) [16], naïve Bayes (NB) [17], support vector machine method (SVM) [17], artificial neural network (ANN) [18] based on the associated document's features.

Our approach improves the accuracy of sentiment and emotion discovery by semantically enriching the metadata from the linked open data and the bibliographic records. This paper presents the design, implementation and evaluation of an enhanced ecosystem, called semantic metadata enrichment ecosystem or SMESE. It includes:

1. An enhanced semantic metadata catalogue.
2. An enhanced harvesting of metadata & data engine.
3. Metadata enrichment based on semantic topic detection and sentiment/emotion analysis.

More specifically, this paper extends our previous work [2] with:

1. SSEA: discovery of sentiments/emotions hidden within the text or linked to a multimedia structure through an AI computational approach.
2. Algorithm for generation of semantic topics by text analysis, relationships and multimedia contents; this second algorithm will be proposed in another paper.

Using simulation, the performance of SSEA was evaluated in terms of accuracy of sentiment and emotion discovery. Existing approaches to enriching metadata, in terms of sentiment and emotion discovery were used for comparison. Simulation results showed that SSEA outperforms existing approaches.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 describes SSEA algorithm. Section 4 presents the evaluation through a number of simulations while Section 5 presents a summary and some suggestions for future work.

II. RELATED WOK

In the past few years, a number of natural language processing (NLP) tasks have been configured for semantic web (SW) tasks including: ontology learning, linked open data, entity resolution, natural language querying to linked data, etc. [19]. This improvement of metadata enrichment using SW involves obtaining hidden data, hence the concept of entity metadata extraction (EME).

Interest in EME was initially limited to those in the SW community who preferred to concentrate on manual design of ontologies as a measure of quality. Following linked data bootstrapping provided by DBpedia, many changes ensued with a consequent need for substantial population of knowledge bases, schema induction from data, natural language access to structured data, and in general all applications that make for joint exploitation of structured and unstructured content. In practice, NLP research started using SW resources as background knowledge. Graph-based methods, meanwhile, were incrementally entering the toolbox of semantic technologies at large.

In the related work section, sentiment and emotion analysis (SEA) that is one field of entity metadata extraction research from text aspect is investigated.

A. Sentiment analysis

The problem of sentiment analysis has been widely studied and different approaches applied, such as machine learning (ML), natural language processing (NLP) and semantic information retrieval (SIR).

There are three main techniques for sentiment analysis [20]:

1. Keyword spotting.
2. Lexical affinity.
3. Statistical methods.

Keyword spotting includes developing a list of keywords that relate to a certain sentiment. These words are usually positive or negative adjectives since such words can be strong indicators of sentiment. Keyword spotting classifies text by affect categories based on the presence of unambiguous affect words such as happy, sad, afraid, and bored.

Lexical affinity is slightly more sophisticated than keyword spotting. Rather than simply detecting obvious affect words, it assigns to arbitrary words a probabilistic 'affinity' for a particular emotion. Lexical affinity determines the polarity of each word using different unsupervised techniques. Next it aggregates the word scores to obtain the polarity score of the text. For example, 'accident' might be assigned a 75% probability of indicating a negative effect, as in 'car accident' or 'injured in an accident'.

Statistical methods, such as Bayesian inference and support vector machines, are supervised approaches in which a labeled corpus is used for training a classification method which builds a classification model used for predicting the polarity of novel texts. By feeding a large training corpus of affectively annotated texts to a machine learning algorithm, it is possible for the system to not only learn the affective valence of affect keywords (as in the keyword spotting approach), but also to take into account the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies. In addition, sophisticated NLP techniques have been developed to address the problems of syntax, negation and irony. Sentiment analysis can be carried out at different levels of text granularity: document [17, 21-25], sentence [1, 4, 6, 26, 27], phrase [28], clause, and word [18, 29, 30].

Sentiment analysis may be at the sentence or phrase level (which has recently received quite a bit of research attention) or at the document level.

From the perspective of this paper, our work may be seen as document-level sentiment analysis—that is, a document is regarded as an opinion on an entity or aspect of it. This level is associated with the task called document-level sentiment classification, i.e., determining whether a document expresses a positive or negative sentiment.

In [8], the authors presented a survey of over one hundred articles published in the last decade on the tasks, approaches, and applications of sentiment analysis. With a major part of available worldwide data being unstructured (such as text, speech, audio, and video), this poses important research challenges. In recent years numerous research efforts have led to automated SEA, an extension of the NLP area of research. The authors identified seven broad classifications:

1. Subjectivity classification
2. Sentiment classification
3. Review usefulness measurement
4. Lexicon creation
5. Opinion word and product aspect extraction
6. Opinion spam detection
7. Various applications of opinion mining

The first five dimensions represent tasks to be performed in the broad area of SEA. For the first three dimensions (subjectivity classification, sentiment classification and review usefulness measurement), the authors note that the applied approaches are broadly classified into three categories:

1. Machine learning
2. Lexicon based
3. Hybrid approaches

Since one of our research objectives was to extract sentiment and emotion metadata from documents, the rest of this section focuses on sentiment classification, lexicon creation, and opinion word and product aspect extraction. Sentiment classification is concerned with determining the polarity of a sentence; that is, whether a sentence is expressing positive, negative or neutral sentiment towards the subject. A lexicon is a vocabulary of sentiment words with respective sentiment polarity and strength value while opinion word and product aspect extraction is used to identify opinion on various parts of a product. As per our research objective the rest of the literature review was oriented to document-level sentiment analysis. For our purposes, we assume that a document expresses sentiments on a single content and is written by a single author.

Cho et al. [23] proposed a method to improve the positive vs. negative classification performance of product reviews by merging, removing, and switching the entry words of the multiple sentiment dictionaries. They merge and revise the entry words of the multiple sentiment lexicons using labeled product reviews. Specifically, they selectively remove the sentiment words from the existing lexicon to prevent erroneous matching of the sentiment words during lexicon-based sentiment classification. Next, they selectively switch the polarity of the sentiment words to adjust the sentiment values to a specific domain. The remove and switch operations are performed using the target domain's labeled data, i.e. online product reviews, by comparing the positive and negative distribution of the labeled reviews with a positive and negative distribution of the sentiment words. They achieved 81.8% accuracy for book reviews. However, their contribution is limited to development of a novel method of removing and switching the content of the existing sentiment lexicons.
Moraes et al. [17] compared popular machine learning approaches (SVM and NB) with an ANN-based method for document-level sentiment classification. Naive Bayes (NB) is a probabilistic learning method that assumes terms occur independently while the support vector machine method (SVM) seeks to maximize the distance to the closest training point from either class in order to achieve better generalization/classification performance on test data. The authors reported that, despite the low computational cost of the NB technique, it was not competitive in terms of classification accuracy when compared to SVM. According to the authors, many researchers have reported that SVM is perhaps the most accurate method for text classification. Artificial neural network (ANN) derives features from linear combinations of the input data and then models the output as a nonlinear function of these features. Experimental results showed that, for book datasets, SVM outperformed ANN when the number of terms exceeded 3,000. Although SVM required less training time, it needed more running time than ANN. For 3,000 terms, ANN required 15 sec training time (with negligible running time) while SVM training time was negligible (1.75 sec). In addition, their contribution was limited to performing comparisons between existing approaches. As in [17], Poria S. et al. [31] experimented with existing approaches and showed that SVM is a better approach for text-based emotion detection.

B. Emotion analysis

This section focuses on sentiment and emotion analysis. Emotions include the interpretation, perception and response to feelings related to the experience of any

particular situation. Emotions are also associated with mood, temperament, personality, outlook and motivation [20, 32, 33]; indeed, the concepts of emotion and sentiment have often been used interchangeably, mostly because both refer to experiences that result from combined biological, cognitive, and social influences. However, sentiments are differentiated from emotions by the duration in which they are experienced. Emotions are brief episodes of brain, autonomic, and behavioral changes. Sentiments have been found to form and be held over a longer period and to be more stable and dispositional than emotions. Moreover, sentiments are formed and directed toward an object, whereas emotions are not always targeted toward an object.

The emotion-topic model (ETM) [34], SWAT model and emotion-term model (ET) [34] are the state-of-the-art models. The SWAT model was proposed to explore the connection between the evoked emotions of readers and news headlines by generating a word-emotion mapping dictionary. For each word *w* in the corpus, it assigns a weight for each emotion *e*, i.e., P(e|w) is the averaged emotion score observed in each news headline *H* in which *w* appears. The emotion-term model is a variant of the NB classifier and was designed to model word-emotion associations. In this model, the probability of word *wj* conditioned on emotion *ek* is estimated based on the co-occurrence count between word *wj* and emotion *ek* for all documents. The emotion-topic model is combination of the emotion-term model and LDA. In this model, the probability of word *wj* conditioned on emotion *ek* is estimated based on the probability of latent topic *z* conditioned on emotion *ek* and the probability of word *wj* conditioned on latent topic *z*.

A number of techniques exist to detect emotions [35]:

1. *Audio based emotion detection*: information from the spectral elements in voice (e.g., speaking rate, pitch, energy of speech, intensity, rhythm regularity, tempo and stress distribution) is used to gather clues about emotions. The features extracted are compared with the training sets in the database using the classifiers.
2. *Blue eyes technology* based on eye moment. In this technique, a picture of the person whose emotions are to be detected is taken and the portion showing his or her eyes is extracted. This extracted image is converted from RGB form to a binary image and compared with ideal eye images depicting various emotions stored in the database. Once the match between the extracted image and one in the database is found, the type of emotion (i.e. happiness, anger, sadness or surprise) is said to be detected.
3. *Facial expression based emotion detection* based on photos of the individual. The images are processed for skin segmentation and analyzed as follows. The image is contrasted, separating the brightest and darkest color in the image area and discriminating the pixels between skin and non-skin. The image is converted into binary form. This processed image is then compared with images forming the training sets in classifiers.
4. *Handwriting based emotion detection* is based on various handwriting indicators or traits of writing (e.g., baseline, slant, pen-pressure, size, zone, strokes, spacing, margins, loops, 'i'-dots, 't'-bar, etc.).
5. *Text based emotion detection* where a computerized NLP approach is used to analyze written text to detect the emotions of the writer. The document is first preprocessed by normalizing the text, then keywords indicating emotional features are extracted. Corresponding emotions are identified through various approaches such as:

a) Keyword spotting technique.
b) Lexical affinity method.
c) Learning based methods.
d) Hybrid method, or by using an emotion ontology which stores a range of emotion classes, associated keywords and relationships.

Text-based emotion detection approaches focus on 'optimistic', 'depressed' and 'irritated.' The limitations are:

1. Ambiguity of keyword definitions.
2. Inability to recognize sentences without keyword.
3. Difficulty determining emotion indicators.

Lei et al. [36] adopted the lexicon-based approach in building the social emotion detection system for online news based on modules of document selection, part-of-speech (POS) tagging, and social emotion lexicon generation. First, they constructed a lexicon in which each word is scored according to multiple emotion labels such as joy, anger, fear, surprise, etc. Next, a

lexicon was used to detect social emotions of news headlines. Specifically, given the training set T and its feature set F, an emotion lexicon is generated as a V×E matrix where the (*j, k*) item in the matrix is the score (probability) of emotion *ek* conditioned on feature *fj*. The authors do not explain how they extracted the features from the document.

Anusha and Sandhya [37] proposed a system for text-based emotion detection which uses a combination of machine learning and natural language processing techniques to recognize affect in the form of six basic emotions proposed by Ekman. They used the Stanford CoreNLP toolkit to create the dependency tree based on word relationships. Next, phrase selection is done using the rules on dependency relationships that gives priority to the semantic information for the classification of a sentence's emotion. Based on the phrase selection, they used the Porter stemming algorithm for stemming, and stopwords removal and tf-idf to build the feature vectors. The authors do not propose a new approach but implement existing algorithms.

Cambria et al. [38] explored how the high generalization performance, low computational complexity, and fast learning speed of extreme learning machines can be exploited to perform analogical reasoning in a vector space model of affective common-sense knowledge. After performing TSVD on AffectNet, they used the Frobenius norm to derive a new matrix. For the emotion categorization model, they used the Duchenne smile and the Klaus Scherer model. As in [37], the authors do not propose a new approach but implement existing algorithms.

III. RESULTS AND DISCUSSION

Table I: Summary of attribute comparison of existing and SSEA algorithm

Existing algorithms	extraction	Classification	Sentiment analysis	Emotion analysis	Concept extraction
AlchemyAPI (http://www.alchemyapi.com/)	x	x	x	x	x
DBpedia Spotlight (https://github.com/dbpedia-spotlight)					x
Wikimeta (https://www.w3.org/2001/sw/wiki/Wikimeta)					x
Yahoo! Content Analysis API (https://developer.yahoo.com/contentanalysis/)		x			x
Open Calais (http://www.opencalais.com/)	x	x			x
Tone Analyzer (https://tone-analyzer-demo.mybluemix.net/)			x	x	
Zemanta (http://www.zemanta.com/)					x
Receptiviti (http://www.receptiviti.ai/)			x	x	
Apache Stanbol (https://stanbol.apache.org/)					x
Bitext (https://www.bitext.com/)			x		x
Mood patrol (https://market.mashape.com/soulhackerslabs/moodpatrol-emotion-detection-from-text)				x	
Aylien (http://aylien.com/)	x	x	x		
AIDA (http://senseable.mit.edu/aida/)					x
Wikifier (http://wikifier.org/)					x
TextRazor (https://www.textrazor.com/)					x
Synesketch (http://krcadinac.com/synesketch/)				x	
Toneapi (http://toneapi.com/)			x	x	
SSEA algorithm	x	x	x	x	x

1. Rule-Based Semantic Metadata Internal Enrichment Engine

This section presents an overview and details of the proposed rule-based semantic metadata internal enrichment engine, including the SSEA algorithm used to process semantic metadata internal enrichment. The main goal of this paper is to enhance the SMESE platform [2] through text analysis approaches for sentiment and emotion and detection.

C. Rule-based semantic metadata internal enrichment engine overview

The rule-based semantic metadata internal enrichment engine has been designed to find short descriptions, in terms of topics, sentiments and emotions of the members of a collection to enable efficient processing of large collections while preserving the semantic and statistical relationships that are useful for tasks such as: topic detection, classification, novelty detection, summarization, and similarity and relevance judgments. Figure 1 shows an overview of the architecture that consists of:

1. User interest-based gateway.
2. Metadata initiatives & concordance rules.
3. Harvesting web metadata & data.
4. User profiling engine.
5. Rule-based semantic metadata internal enrichment engine.

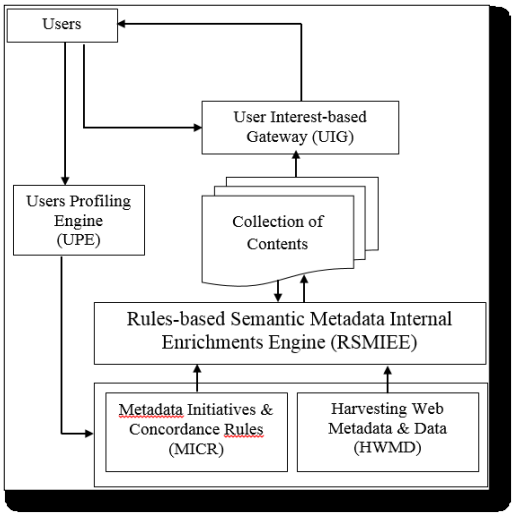

Figure 1: Architecture of the rule-based semantic metadata internal enrichment engine

The user interest-based gateway (UIG) is designed to push notifications to users based on the emotions and interests found using the user-profiling engine. UIG is also a discovery tool that allows users to search and discover contents based on their interests and emotions.
The user-profiling engine applies machine learning algorithms to user feedback in terms of appreciation, rating, comment and historical research in order to provide user profiles. When the contextual information of users is available, it is used to increase the accuracy of the profiling process.

The engine performs automated metadata internal enrichment based on the set of metadata initiatives & concordance rules, the engine for harvesting web metadata & data, the user profile and a thesaurus. This engine implements SSEA for sentiment and emotion detection of documents and an algorithm for topic-automated detection from documents.

SSEA tasks may be redefined as document classification issues as they contain methods for the classification of natural language text. These methods will help to predict the query's category, given a set of training documents with known categories and a new document, which is usually called the query.
The following sub-sections present the terminology and assumptions, the necessary pre-processing and details of the algorithms implemented in the engine.

D. Terminology and assumptions

In this section the following terms are defined:

1. A word or term is the basic unit of discrete data, defined to be an item from a vocabulary indexed by $\{1, \ldots, V\}$. Terms are presented using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the i^{th} term in the vocabulary is represented by an I-vector w such that $w^i = 1$ and $w^j = 0$ for $i \neq j$. For example, let V= {book, image, video, cat, dog} be the vocabulary. The video term is represented by the vector (0, 0, 1, 0, 0).
2. A line is a sequence of N terms denoted by l. These terms are extracted from a real sentence; a sentence is a group of words, usually containing a verb, that expresses a thought in the form of a statement, question, instruction, or exclamation and when written begins with a capital letter.
3. A document is a sequence of N lines denoted by $D = (w_1, w_2; \ldots, w_N)$, where w_i is the i^{th} term in the sequence coming from the lines. D is represented by its lines as $D = (l_1, \ldots l_i, \ldots, l_K)$.
4. A corpus is a collection of M documents denoted by $C = \{D_1, D_2, \ldots, D_M\}$.
5. An emotion word is a word with strong emotional tendency. An emotion word is a probabilistic distribution of emotions and represents a semantically coherent emotion analysis. For example, the word "excitement", presenting a positive and pleased feeling, is assigned a high probability to emotion "joy".

To implement the SSEA algorithm, an initial set of conditions must be established:

1. A list of topics $T = \{t_1, \ldots, t_i, \ldots, t_n\}$ is readily available.
2. Each existing document D_j is already annotated by topic. The annotated topics of document D_j are denoted as $T_{Dj} = \{t_p \ldots, t_i, \ldots, t_q\}$ where t_p, t_i, and $t_q \in T$.
3. The corpus of documents is already classified by topics. $C_{ti}=\{\ldots,D_j,\ldots\}$ denotes the corpus of documents that have been annotated with topic t_i. Note that the document D_j may be located in several corpuses.
4. A list of emotions $E = \{e_1, \ldots, e_i, \ldots, e_E\}$ is readily available with the common instances of e being joy, anger, fear, surprise, touching, empathy, boredom, sadness, warmth.
5. A set of ratings over E emotion labels denoted by $R_{Dj} = \{r_{d,e1} \ldots, r_{d,ei}, \ldots, r_{d,eE}\}$. The value of $r_{d,ei}$ is the number of users who have voted i^{th} emotion label e_i for document d. In other words, $r_{d,ei}$ is the number of

users who claimed that emotion e_i is found in document d.

6. The corpus of documents are already classified by sentiment and emotion based on the user rating. C_{ei} = {…,D_j,…} denotes the corpus of documents rated with emotion e_i. Note that the document D_j may be located in several knowledge corpi.
7. A list of sentiments S = {s_1, … , s_i, … , s_S} is readily available.
8. A thesaurus is available and has a tree hierarchical structure. A thesaurus contains a list of words with synonyms and related concepts. This approach uses synonyms or glosses of lexical resources in order to determine the emotion or polarity of words, sentences and documents.

E. Document Pre-Processing

Before document analysis, SSEA performs a pre-processing. The objective of the pre-processing is to filter noise and adjust the data format to be suitable for the analysis phases. It consists of stemming, phase extraction, part-of-speech filtering and removal of stop-words. The corpus of documents crawled from specific databases or the internet consists of many documents. The documents are pre-processed into a basket dataset C, called document collection. C consists of lines representing the sentences of the documents. Each line consists of terms, i.e. words or phrases. An example of C follows:

C=

……

……

D_j= (line 1) : term 1, term 2, term 6, term 9.
(line 2) : term 10, term 6, term 2, term 3.
……
(line i) : term 3, term 5, term 2, term 3, term 9, term 1.
……
(line N_j) : term 2, term 15, term 9, term 3, term 4.

……

……

More specifically, to obtain D_j, the following preprocessing steps are performed:

1. Language detection.
2. Segmentation: a process of dividing a given document into sentences.
3. Stop word: a process to remove the stop words from the text. Stop words are frequently occurring words such as ‘a’ an’, the’ that provide less meaning and generate noise. Stop words are predefined and stored in an array.
4. Tokenization: separates the input text into separate tokens.
5. Punctuation marks: identifies and treats the spaces and word terminators as the word breaking characters.
6. Word stemming: converts each word into its root form by removing its prefix and suffix for comparison with other words.

More specifically, a standard preprocessing such as tokenization, lowercasing and stemming of all the terms using the Porter stemmer [39]. Therefore, we also parse the texts using the Stanford parser [40] that is a lexicalized probabilistic parser which provides various information such as the syntactic structure of text segments, dependencies and POS tags. ‘Word’ and ‘term’ are used interchangeably in the rest of this paper.

F. Semantic sentiment and emotion analysis: SSEA

The aim of SSEA is to classify the corpus of documents taking emotion into consideration, and to determine which sentiment it more likely belongs to.

A document can be a distribution of emotion $p(e \mid d) e \in E$ and a distribution of sentiment $p(s \mid d) s \in S$. SSEA is a hybrid approach that combines a keyword-based approach and a rule-based approach. SSEA is applied at the basic word level and requires an emotional keyword dictionary that has keywords (emotion words) with corresponding emotion labels.
Next, to refine the detection, SSEA develops various rules to identify emotion. Rules are defined using an affective lexicon that contains a list of lexemes annotated with their affect.

The emotional keyword dictionary and the affective lexicon are implemented in a thesaurus. SSEA is a knowledge-based approach that uses an AI computational technique. The purpose of SSEA is to identify positive and negative opinions and emotions. Figure 2 presents an overview of the architecture of the sentiment and emotion detection process phase.

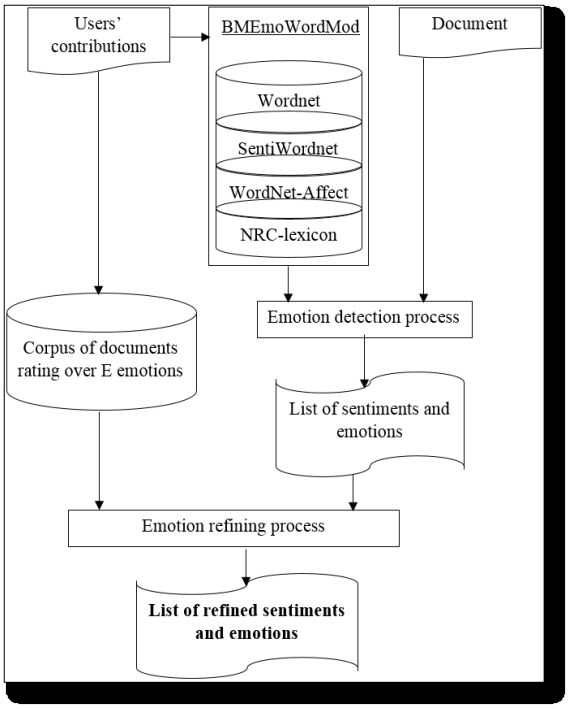

Figure 2: Sentiment and emotion detection process phase – Architecture overview

For affective text evaluation, SSEA uses the SS-Tagger (a part-of-speech tagger) [41] and the Stanford parser [40]. The Stanford parser was selected because it is more tolerant of constructions that are not grammatically correct. This is useful for short sentences such as titles. SSEA also uses several lexical resources that create the SSEA knowledge base located in the thesaurus. The lexical resources used are:

1. WordNet.
2. WordNet-Affect.
3. SentiWordNet.
4. NRC emotion lexicon.

WordNet is a semantic lexicon where words are grouped into sets of synonyms, called synsets. In addition, various semantic relations exist between these synsets (for example: hypernymy and hyponymy, antonymy and derivation). WordNet-Affect is a hierarchy of affective domain labels that can further annotate the synsets representing affective concepts. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity, the sum of which always equals 1.0.

The NRC emotion lexicon is a list of English words and their association with eight basic emotions (anger, anticipation, disgust, fear, joy, sadness, surprise and trust) and two sentiments (negative and positive). The NRC emotion lexicon is a thesaurus that associates for a word, the value one or zero for each emotion. This association is made of binary vectors. The disadvantage of this thesaurus is that since the values are binary, all words belonging to an emotion have the same weight for that emotion. To address this problem, the NRC emotion lexicon thesaurus was combined with the WordNet, WordNet-Affect and SentiWordNet thesaurus. This associates a feelings score with each word-POS. POS1 are grammatical categories used to classify words in dimensions such as adjectives or verbs. SentiWordNet associates with each couple a valence score that can be either negative or positive with respect to the sense of the word in question. The word death, for example, is likely to have a negative score. SSEA also relies on shifter valences. These are lexical expressions capable of changing the valence score of emotions in a text.

For example, take the phrase "I am happy" with a score of 1 for the joy emotion. For the phrase "I am **very** happy", 'very' is a valence intensifier that will change the joy emotion score to 2. In the case, "I am **not** happy" the modifier 'not' will change the emotion joy to the contrary emotion sadness.

The main component of SSEA is the thesaurus, called BM emotion word model (BMEmoWordMod). BMEmoWordMod is an emotion-topic model that provides the emotional score of each keyword by taking the topic into account.

BMEmoWordMod introduces an additional layer (i.e., latent topic) into the emotion-term model such as SentiWordNet. SSEA is composed of three phases:

1. BMEmoWordMod generation process phase.
2. Sentiment and emotion discovery process phase.
3. Sentiment and emotion refining process phase.

The following sub-sections describe the three phases of the SSEA model used to discover sentiment and emotion.

1) BMEmoWordMod generation - process phase

In the first step, a training set from the original corpus is created. The most relevant and discriminative documents are selected automatically. In the second step, each word is tagged with a POS and the combination of word and POS used as the essential feature. Finally,

BMEmoWordMod is generated using the extracted features, which can then be used to discovery the sentiments and emotions of new documents.
Basically, a BMEmoWordMod entry has the following fields:

<Word/POS/synsets_ID><Topics><Emotion_Probability><Sentiment_Probability> where:

1. Emotion_Probability is a vector of ordered emotion label probability such as <anger probability, disgust probability, fear probability, joy probability, sadness probability, surprise probability>.
2. Sentiment_ Probability is a vector of ordered sentiment category probability such as <positive score, negative score>.

For example, the BMEmoWordMod entry for "kill" may look like: <kill/v/00829041><War><0.5, 0.1, 0.3, 0, 0.2, 0><0.1, 0.6>.

Step 1: Training set selection

The objective of this step is to reduce the time for generating the emotion lexicon BMEmoWordMod, while obtaining a better quality lexicon. For each emotion e_i, documents in the corpus are ranked by descending order of ratings over e_i. Next, the emotions with the highest ratings among the documents are chosen. Then relevant documents for a given emotion e_i are selected based on the topic detection algorithm; we assume that this topic detection algorithm is known. The training set selection process terminates when the first phase topic detection algorithm requirements are meet. The training set TS is produced by conducting this step on the entire corpus.

Step 2: Intermediate lexicon generation

Using WordNet-Affect, the WordNet entries are filtered in order to retain only those synsets where the A_labeb is "EMOTION". Then, using SentiWordNet and the NRC emotion lexicon, the sentiment category and emotion value are associated with each selected emotional synset of WordNet. An intermediate lexicon is produced where each entry is <word/POS/synsets_ID><Emotion_value><Sentiment_Score>.
BMEmoWordMod evaluates the probability of each emotion based on the topic and user rating.

Step 3: Sentiment and emotion lexicon generation

The assumption that words in a document are the first indicator of the evoked emotion is assumed to be valid. However, the same word in different contexts may reflect different emotions, and words that bear emotional ambiguity are difficult to recognize out of context. Thus, other strategies are necessary to associate a sentiment or emotion with a given word. The POS of each word is used to alleviate the problem of emotional ambiguity of words and the context dependence of sentiment orientations. The POS of a word is a linguistic category defined by its syntactic or morphological behaviour. Categories include: noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection.

For example, the word ''bear'' has completely different orientations, one positive and one negative, in the following two sentences:

1. Teddy bear: a helping hand for disease sufferers.
2. They have to bear living with a disease.

The word ''bear'' is a noun in the first sentence and a verb in the second. A word feature f_j is defined as the association of the word W_j and its POS, e.g., (Kill/Verb). After defining the word feature f_j, its emotion probability is computed with equation (1):

$$EmoPro\left(e_i \mid f_j, t_k\right) = Val\left(f_j\right) \times \frac{\sum_{d \in C_{tk} \subset ND} p\left(f_j, t_k, d\right) \times oc\left(e_i, t_k\right)}{\sum_{e_l \in E}\left(\sum_{d \in C_{tk} \subset ND} p\left(f_j, t_k, d\right) \times oc\left(e_l, t_k\right)\right)} \quad (1)$$

where:

1. $Val\left(f_j\right)$ denotes the value (1 or 0) of word feature f_j in the intermediate lexicon.
2. $p\left(f_j, t_k, d\right)$ denotes the probability of feature f_j conditioned on document of corpus C_{tk} (subset of documents with topic t_k). $p\left(f_j, t_k, d\right)$ is the number of occurrences of the feature f_j in d divided by the total number of occurrences of all features in d.
3. $oc\left(e_i, t_k\right)$ denotes the co-occurrence number of documents d of C_{tk} and emotion e_i.

This strategy is used to eliminate emotions that are not associated with the same word in the NRC emotion lexicon. The sentiment probability of the word feature f_j is given by equation (2):

$$SenPro(s_i \mid f_j, t_k) = SSco(f_j) \times \frac{\sum_{d \in C_{tk} \subset ND} p(f_j, t_k, d) \times oc(s_i, t_k)}{\sum_{s_l \in S} \left(\sum_{d \in C_{tk} \subset ND} p(f_j, t_k, d) \times oc(s_l, t_k) \right)} \quad (2)$$

where:

1. $SSco(f_j)$ denotes the score of feature f_j in the intermediate lexicon.
2. $oc(s_i, t_k)$ denotes the co-occurrence number of documents d of C_{tk} and sentiment s_i.

Here, s_i may have two values, a positive sentiment S_P and negative sentiment S_N. Finally, to derive BMEmoWordMod, first the topic is added, then the emotion value is replaced by the computed emotion probability and the sentiment score with the computed sentiment probability.

2) Sentiment and emotion discovery - process phase

This phase identifies the sentiments and emotions that are likely associated with a given new document by using the sentiment and emotion semantic lexicon BMEmoWordMod generated in the previous section. After preprocessing, the term vector of the new document is defined using TF-IDF.

Let ND be the new document and $W_{ND} = \{W_1, \ldots, W_z\}$ the set of distinct terms occurring in the corpus of documents. To obtain the z-dimensional term vector that represents each document in the corpus, the tf-idf of each term of W_z is computed. The result of this computation establishes the term vector $\overrightarrow{t_{ND}} = (\text{tfidf}(W_1, ND), \ldots, \text{tfidf}(W_z, ND))$.

Using vector $\overrightarrow{t_{ND}}$, $T_{ND}=\{t_p, \ldots, t_q\}$ obtained using topic detection algorithm (assumed to be known) and BMEmoWordMod, the sentiment and emotion vector of new document $\overrightarrow{E_{f_j, ND}} = (\ E(f_j, \text{ND}, e_1), \ldots, E(f_j, \text{ND}, e_E), E(f_j, \text{ND}, s_P), E(f_j, \text{ND}, s_N)\ )$ is given by equation (3):

$$E(f_j, \text{ND}, e_i) = \frac{\text{tfidf}(W_j, ND)}{\sum_{l=1}^{z} \text{tfidf}(W_l, ND)} \times \sum_{t_k \in T_{ND}} \text{BMEmoWord}(f_j, e_i, t_k) \quad (3)$$

where $\text{BMEmoWord}(f_j, e_i, t_k)$ denotes the emotion probability of emotion e_i for the feature word f_j giving the topic t_k. $\text{BMEmoWord}(f_j, e_i, t_k)$ is selected in BMEmoWordMod.

The weight of emotion e_i for document ND is computed with equation (4):

$$W_E(\text{ND}, e_i) = \sum_{W_j \in W_{ND}} E(f_j, \text{ND}, e_i) \quad (4)$$

Equation (4) yields the emotional vector of new document ND $\overrightarrow{V_{ND}} = (\ W_E(\text{ND}, e_1), \ldots, W_E(\text{ND}, e_i), \ldots, W_E(\text{ND}, e_E), W_E(\text{ND}, s_P), W_E(\text{ND}, s_N)\ )$

Next, the new document ND emotion and sentiment is inferred using a fuzzy logic approach and the emotional vector $\overrightarrow{V_{ND}}$. The weight of emotion is transformed into five linguistic variables: very low, low, medium, high, and very high. Then, using these variables as input to the fuzzy inference system one obtains the final emotion for the new document. The fuzzy logic rules are predefined by experts.

3) Sentiment and emotion refining - process phase

The refining process validates discovered sentiment and emotion after the document analysis. Similarity is computed between new documents and documents in the corpus rated over E emotions. First, the term vectors of each document are defined using the tf-itf of each term, tf-itf is then computed using equation (5); to identify the most important terms of a given document D_j, the tf-idf of each term W_i in the corpus C_{ti} is computed using equation (5) as follows:

$$f(W_i, D_j, C_{ti}) = TF - IDF(W_i, D_j, C_{ti}) = TF(W_i, D_j) * \log\left(\frac{|C_{ti}| = M_i}{IDF(W_i, C_{ti})}\right) \quad (5)$$

Note that the terms extracted from the corpus of documents rated over E emotions are those employed by users. Next, to measure the similarity between two

documents, the cosine similarity of their representative vectors is computed using equation (6); given two documents $\overrightarrow{t_{d1}}$ and $\overrightarrow{t_{d2}}$, their cosine similarity is computed as:

$$SimCos\left(\overrightarrow{t_{d1}}, \overrightarrow{t_{d2}}\right) = \frac{\vec{} \quad \overrightarrow{t}}{\left|\overrightarrow{t_{d1}}\right| \times \left|\overrightarrow{t_{d2}}\right|} \tag{6}$$

Two documents d1 and d2 are similar when the similarity $SimCos(\overrightarrow{t_{d1}}, \overrightarrow{t_{d2}})$ of these two documents is less than the similarity threshold β. Note that it is already assumed that when the similarity $SimCos(\overrightarrow{t_{d1}}, \overrightarrow{t_{d2}})$ of two documents d1 and d2 is less than the similarity threshold β, the documents are not similar.

2. Evaluation using simulations

This section presents an evaluation of SSEA performance using simulations. To perform these simulations, an experimental environment called Libër was used. Libër was developed to provide a simulator to prototype the new algorithm SSEA.

G. Dataset and parameters

To evaluate SSEA, real datasets from different projects that have digital and physical library catalogues were used. These datasets, consisting of 25,000 documents with a vocabulary of 375,000 words, were selected using average TF-IDF for the analysis. The documents covered 20 topics and 8 emotions. The number of documents per topic or emotion was approximately equal. The average number of topics per document was 7 while the average rating emotion number per document was 4. 15,000 documents of the dataset were used for the training phase and the remaining 100 used for the test. Note that the 10,000 documents used for the tests were those that had more annotated topics or a higher rating over emotions.

To measure the performance of topic detection (sentiment and emotion discovery, respectively) approaches, comparison of detected topics (the discovered sentiment and emotion, respectively) with annotation topics of librarian experts (user ratings) were carried out. Table II presents the values of the parameters used in the simulations. The server characteristics for the simulations were: Dell Inc. PowerEdge R630 with 96 Ghz (4 x Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, 10 core and 20 threads per CPU) and 256 GB memory running VMWare ESXi 6.0.

Table II: Simulation Parameters

Parameter	Value
ε	3
NumKeyTerm	8
ω	0.5
β	0.7
λ	0.6
α	100
co-occurrence threshold	0.75
semantic threshold	1
term cluster matching threshold	0.45

H. Performance criteria

SSEA performance was measured in terms of running time [16] and accuracy [42] [43]. Note that in the library domain, the most important criteria was precision while resource consumption was important for the software providers.

The running time, denoted by Rt, was computed as follows:

$$Rt = Et - Bt$$

where *Et* and denotes the time when processing is completed and *Bt* the time when it started.

To compute the accuracy, let E_{rating} and $E_{discovered}$ be the set of rating over emotion and the set of discovered emotion by SSEA for a given document d. The accuracy of sentiment and emotion discovery, denoted by A_d^e, was computed as follows:

$$A_d^e = \frac{2.\left|E_{\text{rating}} \cap E_{\text{discovered}}\right|}{\left|E_{\text{rating}}\right| + \left|E_{\text{discovered}}\right|}$$

Simulation results were averaged over multiple runs with different pseudorandom number generator seeds. The average accuracy, *Ave_acc*, of multiple runs was given by:

$$Ave\_acc = \frac{\sum_{x=1}^{I}\left(\frac{\sum_{d \in TD} A_d^e}{|TD|}\right)}{I}$$

where TD denotes the number of tests documents and I denotes the number of test iterations.

The average running time, *Ave_run_time,* was given by:

$$Ave\_run\_time = \frac{\sum_{x=1}^{I} Rt}{I}$$

I. Sentiment and emotion analysis performance evaluation

SSEA performance was also evaluated in terms of accuracy and running time. Simulations used the dataset and parameters mentioned previously. The performance of SSEA was compared to the approaches described in [34] and [37], referred to as ETM-LDA and AP, respectively. ETM-LDA and AP were selected because they were document-based rather than phrase-based.

1) Comparison of approaches with SSEA

Table III shows the characteristics of the approaches used for comparison with SSEA.

Table III: Sentiment and emotion approaches for comparison

Approach	Granularity	Approach	Training	Refining	Thesaurus	Topic	Emotions
AP [37]	D	L	Y	N	5	N	8
ETM-LDA [34]	D	K	Y	N	6	Y	8
SSEA	C	KR	Y	Y	1,2, 3,4	Y	8

1-WordNet; 2-WordNet-Affect; 3-SentiWordNet; 4-NRC Emotion Lexicon; 5- Stanford CoreNLP; 6-Gibbs sampling; D: Document; C: Configurable as desired; L: Learning based; K: Keyword based; KR: Keyword and Rule based; Y: Yes; N: No

SSEA was the only entirely semantic approach taking into account the rules for inferring emotion. In addition, SSEA used a semantic lexicon. Several approaches used semantic lexicon, but these were limited to phrases rather than documents. The best performance approaches used were AP and ETM_LDA.

2) Results analysis

Figure 3 presents the average running time when varying the number of detected emotions. Training times were excluded because this phase was performed only once. The SSEA training phase took more time than the other approaches due to lexicon aggregation and enrichment by users. The average running time increased with the number of test documents. This is normal, as the larger the number of test documents the longer the average running time to perform the sentiment and emotion discovery.

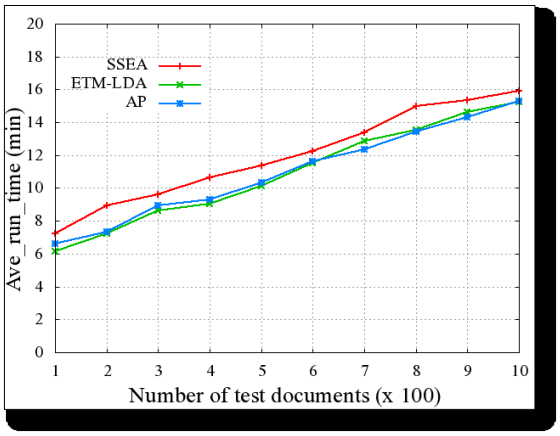

Figure 3: Emotion discovery - Average running time versus number of documents for test phase

Figure 3 shows that ETM-LDA and AP outperformed SSEA on the running time criteria. ETM-LDA required an average of 1.53 sec per document whereas SSEA required an average of 1.74 sec per document. The average relative improvement of ETM-LDA compared with SSEA was approximately 0.21 sec per document. The poorer performance of SSEA resulted from refining sentiment and emotion to increase accuracy.

Figure 4 presents the average accuracy when varying the number of discovered emotions.

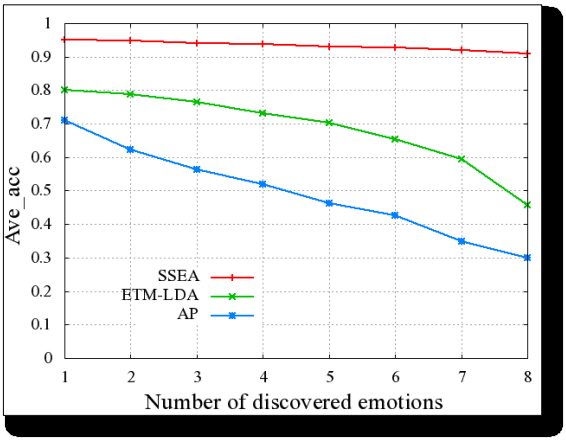

Figure 4: Average detection accuracy for the number of discovered emotions

Positive and negative sentiments were not considered in the accuracy measurement. Figure 4 also shows that the average accuracy decreased with the number of discovered emotions. However, SSEA outperformed the other two approaches used for comparisons. SSEA demonstrated an average accuracy of 93.30% per emotion while ETM-LDA, the best of the other two approaches used for comparison, produced 68.65% accuracy per emotion. The average relative improvement in accuracy of SSEA compared to ETM-LDA was 24.65% per emotion.

In conclusion, the 0.21 sec running time per document increase was, again, a small price to pay for the larger average accuracy of emotion discovery (24.65%).

IV. CONCLUSION

Following is our conclusions on related work in sentiment and emotion analysis:

1. Traditional sentiment analysis methods mainly use terms and their frequency, part of speech, rule of opinions and sentiment shifters. Semantic information is ignored in term selection, and it is difficult to find complete rules.
2. Most of the recent contributions are limited to sentiment analysis elaborated in terms of positive or negative opinion and do not include analysis of emotion.
3. Existing approaches do not take into account the human contribution to improve accuracy.
4. Existing approaches do not combine sentiment and emotion analysis.
5. Lexicon and ontology based approaches provide good accuracy for text-based sentiment and emotion analysis when applying SVM techniques. In our work, it is more important to identify the sentiment and emotion of a book taking into account all the books of the collection. For example, assume that book A has 90% fear and 80% sadness while the emotion which has the best weight of book B is 40% fear; can it be said that fear is the emotion of book B as in book A?
6. Existing approaches do not take into account document collections. In terms of granularity, most of the existing approaches are sentence-based.
7. These approaches do not take into account the context around the sentence and in this way, it is possible to lose the real emotion.

As a general conclusion to the literature review on topic detection, sentiment and emotion analysis, 95% of the work focused on features of the documents (e.g., sentence length, capitalized words, document title, term frequency, and sentences position) to perform text mining and generally make use of existing algorithms or approaches (e.g., LDA, tf-idf, VSM, SVD, LSA, TextRank, PageRank, LexRank, FCA, LTM, SVM, NB and ANN) based on their associated features to documents.

Table I compares the most known text mining algorithms (e.g., AlchemyAPI, DBpedia, Wikimeta, open calais, Bitext, AIDA, TextRazor) with our proposed algorithm in SMESE by keyword extraction, classification, sentiment analysis, emotion analysis and concept extraction.

V. SUMMARY AND FUTURE WORK

In this paper, the goal was to increase the findability (search, discover) of entities based on user interest using external and internal semantic metadata enrichment algorithms. As computers struggle to understand the meaning of natural language, enriching entities semantically with meaningful metadata can improve search engine capability. Words themselves have a wide variety of definitions and interpretations and are often utilized inconsistently. While sentiment and emotion may have no relationship to individual words, thesauri express associative relationships between words, ontologies, entities and a multitude of relationships represented as triplets.

This paper presented an enhanced implementation of SMESE [2] and SSEA algorithm based on text analysis approaches. It includes distinct task that:

1. Discover enriched sentiment and emotion metadata hidden within the text or linked to multimedia structure using the proposed SSEA (Semantic Sentiment and Emotion Analysis) algorithm.
2. Implement rule-based semantic metadata internal enrichment includes algorithm named SSEA.

Table I shows the comparison with most known text mining algorithms (e.g., AlchemyAPI, DBpedia, Wikimeta, Open Calais, Bitext, AIDA, TextRazor) and a new algorithm SSEA with many attributes including keyword extraction, classification, sentiment analysis, emotion analysis, and concept extraction. It was noted

that this algorithm supports more attributes than any other algorithms.

In future work, the focus will be to connect emotion and sentiment to the users evolving interests and will include:

1. Some enhancements to be able to enrich metadata semantically, including the evolution of the user interests over time.
2. Further evaluations of the SSEA model and algorithm with different prototype and datasets.

Exploring text summarization and automatic literature review as metadata enrichments.

VI. REFERENCES

[1]. O. Appel, F. Chiclana, J. Carter, and H. Fujita, "A hybrid approach to the sentiment analysis problem at the sentence level," Knowledge-Based Systems, vol. 108, pp. 110-124, 2016. doi:http://dx.doi.org/10.1016/j.knosys.2016.05.040

[2]. R. Brisebois, A. Abran, and A. Nadembega, "A Semantic Metadata Enrichment Software Ecosystem (SMESE) based on a Multi-platform Metadata Model for Digital Libraries," Accepted for publication in Journal of Software Engineering and Applications (JSEA), vol. 10, no. 04, 2017

[3]. G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1988. doi:http://dx.doi.org/10.1016/0306-4573(88)90021-0

[4]. T. Niu, S. Zhu, L. Pang, and A. El Saddik, "Sentiment Analysis on Multi-View Social Data," in 22nd International Conference on MultiMedia Modeling (MMM), Miami, FL, USA, 2016, pp. 15-27. doi:http://dx.doi.org/10.1007/978-3-319-27674-8_2

[5]. K. Bougiatiotis, and T. Giannakopoulos, "Content Representation and Similarity of Movies based on Topic Extraction from Subtitles," in Proceedings of the 9th Hellenic Conference on Artificial Intelligence, Thessaloniki, Greece, 2016, pp. 1-7. doi:http://dx.doi.org/10.1145/2903220.2903235

[6]. G. A. Patel, and N. Madia, "A Survey: Ontology Based Information Retrieval For Sentiment Analysis," International Journal of Scientific Research in Science, Engineering and Technology, vol. 2, no. 2, pp. 460-465, 2016

[7]. J. A. Balazs, and J. D. Velásquez, "Opinion Mining and Information Fusion: A survey," Information Fusion, vol. 27, pp. 95-110, 2016. doi:http://dx.doi.org/10.1016/j.inffus.2015.06.002

[8]. K. Ravi, and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," Knowledge-Based Systems, vol. 89, pp. 14-46, 2015. doi:http://dx.doi.org/10.1016/j.knosys.2015.06.015

[9]. J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, vol. 311, pp. 18-38, 2015. doi:http://dx.doi.org/10.1016/j.ins.2015.03.040

[10]. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," Comput. Linguist., vol. 37, no. 2, pp. 267-307, 2011. doi:10.1162/COLI_a_00049

[11]. D. Vilares, M. A. Alonso, and C. GÓMez-RodrÍGuez, "A syntactic approach for opinion mining on Spanish reviews," Natural Language Engineering, vol. 21, no. 1, pp. 139-163, 2015. doi:http://dx.doi.org/10.1017/S1351324913000181

[12]. S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," Journal of Artificial Intelligence Research, vol. 50, no. 1, pp. 723-762, 2014. doi:http://dx.doi.org/10.1613/jair.4272

[13]. D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003

[14]. S. T. Dumais, "Latent semantic analysis," Annual Review of Information Science and Technology, vol. 38, no. 1, pp. 188-230, 2004. doi:10.1002/aris.1440380105

[15]. J. Cigarrán, Á. Castellanos, and A. García-Serrano, "A step forward for Topic Detection in Twitter: An FCA-based approach," Expert Systems with Applications, vol. 57, pp. 21-36, 2016. doi:http://dx.doi.org/10.1016/j.eswa.2016.03.011

[16]. P. Chen, N. L. Zhang, T. Liu, L. K. M. Poon, and Z. Chen, "Latent Tree Models for Hierarchical Topic Detection," arXiv preprint arXiv:1605.06650 cs.CL], pp. 1-44, 2016

[17]. R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," Expert Systems with Applications, vol. 40, no. 2, pp. 621-633, 2013. doi:http://dx.doi.org/10.1016/j.eswa.2012.07.059

[18]. M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network," Expert Systems with Applications, vol. 40, no. 16, pp. 6266-6282, 2013. doi:http://dx.doi.org/10.1016/j.eswa.2013.05.057

[19]. A. Gangemi, "A Comparison of Knowledge Extraction Tools for the Semantic Web," in 10th European Semantic Web Conference (ESWC), Montpellier, France, 2013, pp. 351-366. doi:http://dx.doi.org/10.1007/978-3-642-38288-8_24

[20]. S. N. Shivhare, and S. Khethawat, "Emotion Detection from Text," in Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Delhi, India, 2012, pp. 1-7

[21]. A. Moreo, M. Romero, J. L. Castro, and J. M. Zurita, "Lexicon-based Comments-oriented News Sentiment Analyzer system," Expert Systems with Applications, vol. 39, no. 10, pp. 9166-9180, 2012. doi:http://dx.doi.org/10.1016/j.eswa.2012.02.057

[22]. C. Bosco, V. Patti, and A. Bolioli, "Developing corpora for sentiment analysis: The case of irony and senti-tut," IEEE Intelligent Systems, vol. 28, no. 2, pp. 55-63, 2013

[23]. H. Cho, S. Kim, J. Lee, and J.-S. Lee, "Data-driven integration of multiple sentiment dictionaries for lexicon-based sentiment classification of product reviews," Knowledge-Based Systems, vol. 71, pp. 61-71, 2014. doi:http://dx.doi.org/10.1016/j.knosys.2014.06.001

[24]. C. Lin, Y. He, R. Everson, and S. Ruger, "Weakly Supervised Joint Sentiment-Topic Detection from Text," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 6, pp. 1134-1145, 2012. doi:http://dx.doi.org/10.1109/TKDE.2011.48

[25]. E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades, "Ontology-based sentiment analysis of twitter posts," Expert Systems with Applications, vol. 40, no. 10, pp. 4065-4074, 2013. doi:http://dx.doi.org/10.1016/j.eswa.2013.01.001

[26]. B. Desmet, and V. Hoste, "Emotion detection in suicide notes," Expert Systems with Applications, vol. 40, no. 16, pp. 6351-6358, 2013. doi:http://dx.doi.org/10.1016/j.eswa.2013.05.050

[27]. M. Abdul-Mageed, M. Diab, and S. Kübler, "SAMAR: Subjectivity and sentiment analysis for Arabic social media," Computer Speech & Language, vol. 28, no. 1, pp. 20-37, 2014. doi:http://dx.doi.org/10.1016/j.csl.2013.03.001

[28]. L. K.-W. Tan, J.-C. Na, Y.-L. Theng, and K. Chang, "Phrase-Level Sentiment Polarity Classification Using Rule-Based Typed Dependencies and Additional Complex Phrases Consideration," Journal of Computer Science and Technology, vol. 27, no. 3, pp. 650-666, 2012. doi:http://dx.doi.org/10.1007/s11390-012-1251-y

[29]. L. Chen, L. Qi, and F. Wang, "Comparison of feature-level learning methods for mining online consumer reviews," Expert Systems with Applications, vol. 39, no. 10, pp. 9588-9601, 2012. doi:http://dx.doi.org/10.1016/j.eswa.2012.02.158

[30]. C. Quan, and F. Ren, "Unsupervised product feature extraction for feature-oriented opinion determination," Information Sciences, vol. 272, pp. 16-28, 2014. doi:http://dx.doi.org/10.1016/j.ins.2014.02.063

[31]. S. Poria, E. Cambria, A. Hussain, and G.-B. Huang, "Towards an intelligent framework for multimodal affective data analysis," Neural Networks, vol. 63, pp. 104-116, 2015. doi:http://dx.doi.org/10.1016/j.neunet.2014.10.005

[32]. M. D. Munezero, C. S. Montero, E. Sutinen, and J. Pajunen, "Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text," IEEE Transactions on Affective Computing, vol. 5, no. 2, pp. 101-111, 2014. doi:http://dx.doi.org/10.1109/TAFFC.2014.2317187

[33]. W. Li, and H. Xu, "Text-based emotion classification using emotion cause extraction," Expert Systems with Applications, vol. 41, no. 4, Part 2, pp. 1742-1749, 2014. doi:http://dx.doi.org/10.1016/j.eswa.2013.08.073

[34]. S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han, and Y. Yu, "Mining Social Emotions from Affective Text," IEEE Transactions on

Knowledge and Data Engineering, vol. 24, no. 9, pp. 1658-1670, 2012. doi:http://dx.doi.org/10.1109/TKDE.2011.188

[35]. S. V. Kedar, D. S. Bormane, A. Dhadwal, S. Alone, and R. Agarwal, "Automatic Emotion Recognition through Handwriting Analysis: A Review," in 2015 International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2015, pp. 811-816. doi:http://dx.doi.org/10.1109/ICCUBEA.2015.162

[36]. J. Lei, Y. Rao, Q. Li, X. Quan, and L. Wenyin, "Towards building a social emotion detection system for online news," Future Generation Computer Systems, vol. 37, pp. 438-448, 2014. doi:http://dx.doi.org/10.1016/j.future.2013.09.024

[37]. V. Anusha, and B. Sandhya, "A Learning Based Emotion Classifier with Semantic Text Processing," Advances in Intelligent Informatics, M. E.-S. El-Alfy, M. S. Thampi, H. Takagi, S. Piramuthu and T. Hanne, eds., pp. 371-382, Cham, Switzerland: Springer International Publishing, 2015. doi:http://dx.doi.org/10.1007/978-3-319-11218-3_34

[38]. E. Cambria, P. Gastaldo, F. Bisio, and R. Zunino, "An ELM-based model for affective analogical reasoning," Neurocomputing, vol. 149, Part A, pp. 443-455, 2015. doi:http://dx.doi.org/10.1016/j.neucom.2014.01.064

[39]. M. F. Porter, "An algorithm for suffix stripping," Program, vol. 14, no. 3, pp. 130-137, 1980. doi:doi:10.1108/eb046814

[40]. de Marneffe M-C, MacCartney B, and Manning CD, "Generating typed dependency parsers from phrase structure parses " in fifth international conference on language resources and evaluation, GENOA , ITALY 2006, pp. 449–54

[41]. Y. Tsuruoka, and J. i. Tsujii, "Bidirectional inference with the easiest-first strategy for tagging sequence data," in Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, 2005, pp. 467-474. doi:10.3115/1220575.1220634

[42]. C. Zhang, H. Wang, L. Cao, W. Wang, and F. Xu, "A hybrid term–term relations analysis approach for topic detection," Knowledge-Based Systems, vol. 93, pp. 109-120, 2016. doi:http://dx.doi.org/10.1016/j.knosys.2015.11.006

[43]. H. Sayyadi, and L. Raschid, "A Graph Analytical Approach for Topic Detection," ACM Transactions on Internet Technology, vol. 13, no. 2, pp. 1-23, 2013. doi:http://dx.doi.org/10.1145/2542214.2542215