Difficulties and Improvements to Graph-based Lexical Sentiment Analysis using LISA

Mireille Fares[1], Angela Moufarrej[1], Eliane Jreij[1], Joe Tekli[1], and William Grosky[2]
[1] Lebanese American University (LAU), ECE Dept., Byblos, Lebanon
[2] University of Michigan (UM), CIS Dept., Dearborn, USA

Abstract*—Lexical sentiment analysis (LSA) underlines a family of methods combining natural language processing, machine learning, or graph navigation techniques to identify the underlying sentiments or emotions carried in textual data. In this paper, we introduce LISA, an unsupervised word-level knowledge graph-based LexIcal Sentiment Analysis framework. It uses different variants of shortest path graph navigation techniques to compute and propagate affective scores in a lexical-affective graph (LAG), created by connecting a typical lexical knowledgebase (KB) like WordNet, with a reliable affect KB like WordNet-Affect Hierarchy. LISA was designed in two consecutive iterations, producing two main modules: i) LISA 1.0 for affect navigation, and ii) LISA 2.0 for affect propagation and lookup. LISA 1.0 suffered from the semantic connectivity problem shared by some existing lexicon-based methods, and required polynomial execution time. This led to the development of LISA 2.0, which i) processes affective relationships separately from lexical/semantic connections (solving the semantic connectivity problem of LISA 1.0), and ii) produces a *sentiment lexicon* which can be searched in logarithmic time (handling LISA 1.0's efficiency problem). Experimental results on the ANEW dataset show that LISA 2.0, while completely unsupervised, is on a par with existing supervised solutions, highlighting its quality and potential.

***Keywords*—Sentiment Analysis, Affect Analysis, Knowledge Base, Graph Navigation, Sentiment Lexicon, ANEW.**

I. INTRODUCTION

Lexical sentiment analysis (or LSA) systems are automated tools which analyze words and text extracts provided by users, and attempt to classify them under different sentiment categories, such as: *positive*, *negative*, or *neutral* emotions. Affect analysis is a more fine-grained approach of LSA, involving more specific classes of affective emotions such as: *happiness*, *sadness*, *surprise*, and *anger*, etc. LSA is becoming increasingly popular in a wide range of Web applications covering: blog sentiment analysis [15, 84] (in web forums), client feedback analysis [19, 75] (customer opinions on products), sentiment analysis on social media [41, 73] (analyzing tweets or posts on social media), and therapeutic and social emotion analysis [17, 51] (helping autistic children express their emotions).

Most existing LSA approaches (cf. Section II) have utilized *supervised* learning techniques applied on *corpus-based* statistics in order to match words or textual patterns with sentiments represented as labeled categories, e.g., [23, 43]. They usually require extensive training data, training time, and large statistical corpora which are not always available and require significant manual effort. In addition, most methods usually produce discrete sentiment labels (e.g., *joy*, *surprise*) without however evaluating sentiment intensity (valence) scores (e.g., 20% *joy*, 35% *surprise*), e.g., [10, 49]. On the other hand, other studies have utilized *unsupervised* and *lexicon-based* approaches, e.g., [22, 86], to match target words with seed words in a *sentiment lexicon* (e.g., LEW list [20], or WNA list [76]), by evaluating their semantic similarity or distance in a reference lexical knowledge base (KB, e.g., WordNet [48]). The latter usually suffer from the limited coverage of manually created sentiment lexicons, and the limited or inconsistent connectivity of affective concepts in the lexical KB (cf. Section II). Recent efforts have focused on the automatic creation of sentiment corpora, e.g., [4, 7, 57]. Yet most rely on supervised processes for their construction, thus sharing the limitations of supervised method mentioned above.

In this study, we introduce LISA, a framework for unsupervised word-level graph-based LexIcal Sentiment Analysis. Our approach utilizes graph navigation techniques applied on a Lexical-Affective Graph (LAG), to infer word affect scores. The LAG is created by connecting a typical lexical KB graph like WordNet, with a reliable and comprehensive affect KB like WordNet-Affect Hierarchy (WNAH) [65]. LISA was designed in two consecutive iterations, producing two main modules: i) LISA 1.0 for affect navigation, and ii) LISA 2.0 for affect propagation and lookup. LISA 1.0 suffered from the semantic connectivity problem shared by some existing lexicon-based methods, and required polynomial execution time. This led to the development of LISA 2.0, which i) processes affective relationships separately from lexical/semantic connections (solving the semantic connectivity problem of LISA 1.0), and ii) produces a *sentiment lexicon* which can be searched in logarithmic time (handling LISA 1.0's efficiency problem). We have implemented LISA 1.0 and 2.0 to test and evaluate their performance. Results on the Affective Norms for English Words (ANEW) dataset [5, 63] show that LISA 2.0, while completely unsupervised, is on a par with existing supervised solutions, highlighting its quality and potential.

The remainder of the paper is organized as follows. Section II reviews the literature on LSA techniques. Our LISA framework is developed in Section III. Section IV presents experimental results, while Section V concludes with future directions.

II. LITERATURE REVIEW

LSA methods can be described and distinguished following a number of criteria, including: sentiment categories, text granularity, textual features, external resources, and the computation techniques used.

Sentiment categories: Researchers in LSA usually distinguish between two kinds of sentiments: i) opinions/polarity such as *like*/*dislike*, referred to as *positive*/*negative* opinions, and ii) emotions/feelings such as *happy*/*angry*/*afraid*/etc., referred to as affect categories [31]. Accordingly, LSA methods can be distinguished as: i) *opinion detection* (or *opinion mining*) methods [28], and ii) *affect analysis* methods [66]. Affect analysis involves a larger number of affect classes, ranging from a reduced set of six basic emotions in [54] (i.e., *anger*, *fear*, *joy*, *love*, *sadness*, and *surprise*) to a comprehensive hierarchy of 294 sentiment categories introduced in WNAH [65] (cf. Section III.A).

Granularity of LSA: Sentiments can be extracted at different text granularity levels: i) word, ii) phrase, iii) sentence, iv) document, and v) aspect. *Word-level* LSA is the most fine-grained approach where individual words are associated with sentiment categories [1, 59]. *Phrase-level* LSA consists of associating sentiments with individual phrases, where a phrase designates an expression made of a couple of adjacent words (e.g., "unpredictable steering") where the phrase sentiments are deduced from word-level sentiments [82, 85]. Similarly for *sentence-level* and *document-level* LSA, allowing to associate sentiments with individual sentences/documents, based on word-level, phrase-level, or sentence-level LSA [3, 21]. *Aspect-level* LSA consists in extracting the main aspects of a text where aspects represent interesting features describing what the text is about (e.g., "battery", "processor", "touch screen" could be aspects describing *mobile phones*), and then estimating the sentiment scores of the text per aspect [2, 61]. In our current study, we focus on word-level LSA.

Features for LSA: Different features can be utilized to perform word-level LSA, including: i) lexical form, ii) semantic meaning, and iii) part-of-speech tag. Words targeted for LSA are usually matched against a set of seed words with associated sentiments, to acquire/inherit the corresponding sentiment categories [25, 27]. The *part-of-speech* (POS) feature allows distinguishing between nouns, verbs, adjectives, and adverbs which might carry slightly different sentiment clues [34, 79]. The *semantic meaning* feature allows matching words based on their meanings, by comparing their semantic definitions and relationships w.r.t. a lexical KB like WordNet [7, 38]. Other features include: n-gram (word associations) [1, 50], syntactic structure (parse tree) [82, 87], valence shifters (e.g., "really", "could" and "should") [27, 87], and statistical features (e.g., contextual and co-occurrence frequencies) [44, 61]. In our study, we target word-level LSA and thus focus on word-level features.

Resources for LSA: External resources provide reference data which is needed to associate sentiments with text. Here, LSA methods can be distinguished as: i) corpus-based or ii) lexicon-based. The *corpus-based* approach, e.g., [44, 80], is data-driven, as it relies on processing large text corpora (such as OpenMind [62] and ISEAR [60]) to identify the probability of occurrence of textual features, in order to enable sentiment predictions for new texts. The *lexicon-based* approach, e.g., [7, 39], is knowledge-driven, as it relies on acquiring sentiment clues from a readily available sentiment lexicon, i.e., a large collection of words or concepts (i.e., word senses) associated with sentiment categories. Machine readable lexicons such as SentiWordNet [4], WNA [76], and SenticNet [8] are few of the most widely used sentiment lexicons in the literature. While corpus-based methods have been popular in the past few years [24, 42], yet they are generally data hungry and require extensive training, huge textual corpora, and a considerable amount of manual effort which are not always available or feasible in practice.

Yet, lexicon-based LSA methods suffer in turn from two major limitations: i) ambiguity and ii) limited coverage [13, 53]. On the one hand, many widely used sentiment lexicons (such as General Inquirer [64] and LIWC [55]) associate sentiments with words instead of concepts (i.e., word meanings), and thus do not distinguish between the different meanings of the same word which might have – each – a different sentiment bearing. On the other hand, the limited coverage of manually created lexicons (such as the LEW list [20] and the core WNA list [76]) is another major concern, due to the substantial effort in manually annotating terms or concepts [13]. In our study, we focus on lexicon-based LSA, and address both: i) the ambiguity problem by using unambiguous word meanings (concepts) to perform LSA, and ii) the limited coverage problem by connecting a comprehensive affect KB (WNAH) with an expressive lexical KB (WordNet)[1].

Techniques for LSA: Existing LSA approaches can be roughly categorized as: i) supervised, or ii) unsupervised. *Supervised* methods, e.g., [12, 32, 33], involve the use of supervised-learning techniques, using manually annotated samples words/phrases provided as training data for a learning algorithm that induces rules to be used for assigning sentiments with other occurrences of the words/phrases. External knowledge (mainly corpus-based) is used and combined with the human expert's own knowledge of word/phrase sentiments when manually annotating the training examples. Here, different kinds of classifiers have been used, including Support Vector Machines (SVM) [10, 49], Naïve Bayes (NB) [37, 81], Maximum Entropy (ME) [46, 58], and Linear Regression [27, 83]. While effective, supervised methods suffer from several disadvantages. First, they include a learning phase which is time-consuming and subject to over-fitting, depending on the training data set which is not always available. Another shortcoming is that legacy supervised classifiers can only deal with discrete class labels (e.g., *positive*, *calm*, etc.), whereas sentiment intensity (valence) can vary along a continuum (e.g., 80% *positive*, 20% *calm*, etc.). A third shortcoming is that supervised methods train their classifiers to recognize different classes separately, as if the produced categories are totally unrelated, e.g., [10, 49]. Yet, certain sentiment classes may be related [66] (cf. Section III.A). For instance, *hate* and *anger* are related affects and usually co-occur together.

Unsupervised methods, e.g., [22, 78, 86], are usually fully automated and do not require human intervention or a training phase. Most approaches in this category make use of a machine-readable sentiment lexicon (e.g., SentiWordNet [4] or WNA [76]) usually represented as a set of words/expressions or concepts with their sentiment categories or intensity scores. Given a target text to be processed, unsupervised LSA consists in assigning each constituent textual token (e.g., word or phrase) and consequently the whole target text, with a sentiment score. The score is a measurement of the intensity of the token w.r.t. to one (or many) sentiment category(ies). Scoring methods can be distinguished as: i) statistical, or ii) semantic. *Statistical scoring* methods evaluate word average sentiment intensities across the lexicon's items occurring in a text [45, 66]. They assess the intensity of each word based on its co-occurrence frequency with a set of core words reflective of a given affect [14, 52, 74]. The main limitation of this group of methods is the need for a large and expressive textual corpus to perform statistical analysis. *Semantic scoring* consists in evaluating the semantic distance between the meanings of words in a reference KB [59]. Most semantic scoring LSA methods, e.g., [13, 36], utilize WordNet [48] as a widely used lexical KB made of a set of word concepts (synsets) and their semantic relationships (e.g., *synonymy*, *hyponymy*, etc., [9, 67], cf. Section III). In this context, the authors in [40] expand the seed words associated with an affect category by comparing each candidate word and its synonymous terms with the seed word list [49]. In [36], the authors identify the polarity of an input (source) word by measuring its distance in number of *synonymy* relationships (links) from two reference (destination) concepts: *good* and *bad* in the WordNet graph. Similar approaches were introduced in [13, 26], which consider a set of seed concepts (instead of two concepts only: *good* and *bad*) as references for their distance computations. Note that applying the semantic scoring LSA approach requires *word sense disambiguation* (WSD) [68, 69], a computationally expensive pre-processing step to assign the word targeted for LSA with its semantic concept (meaning) [47], so that the latter concept can then be processed for semantic scoring. Another common pitfall of this category of methods is the semantic connectivity between reference concepts, which might not be accurate. For instance, one can traverse the WordNet graph from concepts *good* to *bad* in only three hops using the *synonymy* relationship (cf. Fig. 1). This seems "weird" since *good* and *bad* are opposing sentiments, and one tends to think they should be farther away from each other. This problem is shared among other lexical knowledge references such as ConceptNet [72] and Yago [30], where concepts are defined following their lexical meanings, rather than their affective expressiveness.

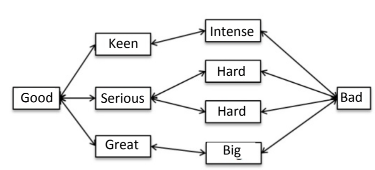

Fig. 1. Extract of *synonymy* relationship connectivity between words *good* and *bad* in WordNet [26]

[1] Other lexical and affective KBs sharing similar properties could also be used.

In this context, there is a crucial need to distinguish between lexical and semantic relationships between concepts in the lexical knowledge graph on the one hand, and affective relationships between affect categories on the other hand.

III. *LISA* FRAMEWORK

To address most of the limitations above, we introduce LISA, an unsupervised word-level knowledge graph-based LexIcal Sentiment Analysis framework. It uses different variants of shortest path graph navigation techniques to compute and propagate affective scores in a Lexical-Affective Graph (LAG). LISA's overall architecture is depicted in Fig. 2. It is designed in two separate yet interconnected modules: LISA 1.0 for affect navigation, and LISA 2.0 for affect propagation and lookup, described in the following subsections.

A. *Lexical Affective Graph*

The Lexical-Affective Graph (or LAG) is created by connecting a typical lexical KB graph like WordNet [48], with a reliable affect reference like WordNet-Affect Hierarchy (WNAH) [65] (although any other lexical or affective references sharing similar properties can be utilized). To our knowledge, WNAH is the most comprehensive affect hierarchy to date, consisting of 294 different affect categories (e.g., *positive emotion*, *joy*, *love*, *apathy*, *euphoria*, etc.), hierarchically organized following a hypernymy/hyponymy (*IsA*/*HasA*) inheritance structure, where every affect category matches a lexical concept (synset) in WordNet. A sample LAG extract is shown in Fig. 3. Word concepts (synsets) in WordNet matching affect categories from WNAH are highlighted with thick contours. *Hypernymy/hyponymy* relationships connecting affect concepts from WNAH are highlighted, to distinguish them from WordNet lexical relationships (which are labeled in the LAG).

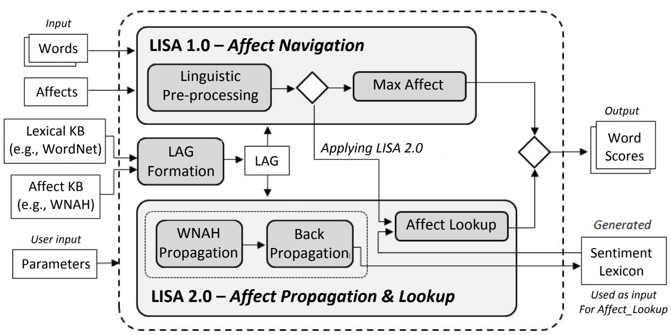

Fig. 2. Simplified activity diagram describing LISA's architecture

B. *LISA 1.0 – Affect Navigation*

The LISA 1.0 affect navigation module accepts as input a set of user words and a set of target affect categories in a LAG, and produces as output the target affect scores (intensity weights) for every input word located in the LAG. It consists of two main components: i) *linguistic pre-processing*, to process input words, identifying their proper word meanings (concepts) in the LAG, and ii) *Max_Affect* which navigates the LAG from the input word concepts to the target affect categories, using an adaptation of the shortest path problem.

1. Linguistic Pre-Processing component

Linguistic pre-processing consists of four main phases: i) *tokenization*, ii) *stop word removal*, iii) *stemming*, and iv) *word sense disambiguation* (WSD). Once located in the LAG, word concepts are provided as input to *Max_Affect* to compute their affective weights.

Note that we utilize the well known *simplified LESK* algorithm [35] to perform WSD, which compares the target word's context (its surrounding words) with the contexts of its different possible meanings (concepts) in the lexical KB, and chooses the concept whose context is most similar to the target word context as its proper (disambiguated) meaning [35]. *Simplified LESK* is one of the most efficient WSD algorithms [77], requiring linear time w.r.t. the number of meanings for a given word, and their context sizes.

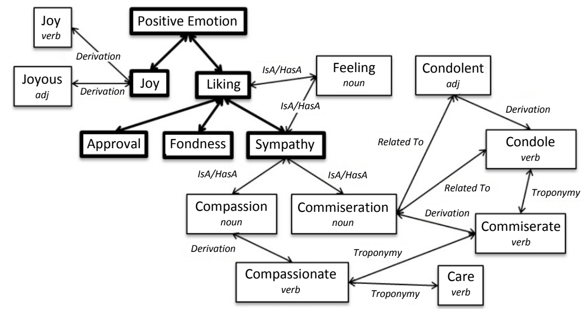

Fig. 3. Sample LAG based on a mapping of WordNet and WNAH

2. Max_Affect Component

The *Max_Affect* component's pseudo-code is described in Fig. 4. It accepts as input the users' disambiguated word concepts and their target affective categories (i.e., the emotions they are interested in), and then produces as output the corresponding sentiment scores in the form of sentiment vectors whose dimensions correspond to the user-chosen affective categories. It utilizes an adaptation of *Dijkstra*'s shortest path distance computations [11], applied on the LAG.

Max_Affect explores the LAG starting from one or multiple word concept nodes. From every starting concept node, it attempts to identify the closest path to every target affective category node, highlighting the target affect's expressiveness w.r.t. the source concept(s). Yet, we altered *Dijkstra*'s original premise: instead of identifying the minimum weight path, *Max_Affect* seeks to identify the maximum sentiment weight of a source concept node c_i w.r.t. a target affect node a_j. We compute node and edge weights as follows:

i. The weight of a source concept node c_i w.r.t. a target affect node a_j, noted $w(c_i, a_j)$ or $w_{aj}(c_i)$, is $\in$ [0, 1], where 0 means that affect category a_j is not expressed in c_i, whereas 1 means that a_j is totally expressed in c_i. The weight of c_i w.r.t. a set of target affect categories $A=\{a_1,\ldots,a_J\}$, consists of a vector of affect weights $V_i = < w(c_i, a_1), \ldots, w(c_i, a_J) >$, of J dimensions, where dimension j corresponds to a target affect category $a_j \in A$, and its vector coordinate $w(c_i, a_j)$ represents the affective weight of a_j w.r.t. c_i.
ii. The weight of an edge outgoing from node c_i and incoming into node c_r, noted $w(c_i, c_r)$, is $\in$ [0, 1] and reflects sentiment "conductance" where 0 means that the edge does not carry any sentiment expressiveness from c_i to c_j, whereas 1 means that the edge carries all the sentiment expressiveness from c_i to c_j. The edge weight is determined firstly based on the edge label (i.e., semantic relationship connecting the two nodes, e.g., *hypernymy*, *related to*, etc.), and secondly based on the out-degree of c_i (depending on the semantic relationship being processed):

$$w(c_i, c_r) = \begin{cases} \dfrac{1}{out\text{-}degree_{rel}(\mathrm{c_i})} & \text{if } rel \notin R_{reliable} \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

where rel designates the edge's label (semantic relationship), and $R_{reliable}$ the set of sentiment reliable relationships. In other words, $w(c_i, c_r) = 1$ (maximum score) if its edge label corresponds to a sentiment reliable relationship, otherwise, it is determined by the out-degree of incoming node c_i. The rationale is that an edge designates a stronger connection between two (word concept or affective category) nodes when it carries most of the descriptive power from the source to the destination, such that the source node has few other out-going connections (if any, cf. Fig. 4).

iii. Finally, instead of starting from an initial weight =0 assigned to the source lexical node c_i, *Max_Affect* starts with an initial weight =1 (maximum sentiment expressiveness), and then multiplies (instead of summing) the source node's weight by the weights of every edge on the maximum weight path leading to a_j. If all edges on the path between c_i and a_j are of maximum sentiment conductance (i.e., they carry all of the sentiment expressiveness), then $w(c_i, a_j) = w_{aj}(c_i) = 1$ where affect a_j is fully expressed in c_i. Otherwise, if edges have diminishing sentiment conductance, then $w_{aj}(c_i)$ will decrease accordingly.

```
Algorithm: Max_Affect
Input: LAG graph: G
       Set of source word concept nodes: C
       Set of destination affect nodes: A
Ouput: Set of affect vectors: ∇
Begin
  Initialize ∇ = {Vi} i = 1... |C|  where  Vi = <w(ci, a1)=0,..., w(ci, a|A|)=0>
  Initialize weights of nodes ∈ G to 0
  Processed = ∅
  For every ci in C
  {
     Frontier = ci, Explored = ∅
     Initialize w(ci) =1, remove from Frontier and add to Explored
     While (Explored ≠ A)
     {
        For each node cj ∈ Explored
          For each node cm outgoing from cj
             Add cm to Frontier
             Compute weight vector of cm
                 w(cm) = max(w(cm), w(cm, cj) × w(cj))
          Compute maximum weight wmax for all nodes in Frontier
          For each node cn in Frontier having w(cn) == wmax
             If cn ∈ Processed Then
                Compute Vi = Vn × w(cn, ci)
                Add ci to Processed
                Goto Exit
             Else Remove cn from Frontier and add to Explored
     }
     Exit:
     If (ci ∉ Processed) Then
        Compute Vi = <w(a1), …, w(a|A|)>
        Add ci to Processed
  }
  Return ∇
End
```

Fig. 4. Pseudo-code of algorithm *Max_Affect*

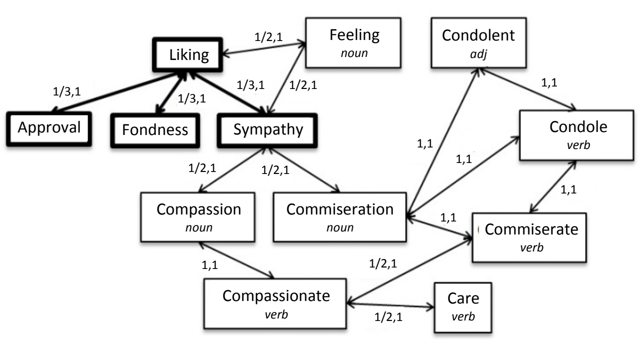

Fig. 5. Sample LAG (lexical affective graph) from Fig. 3, with affect concepts highlighted in bold, to distinguish them from lexical concepts

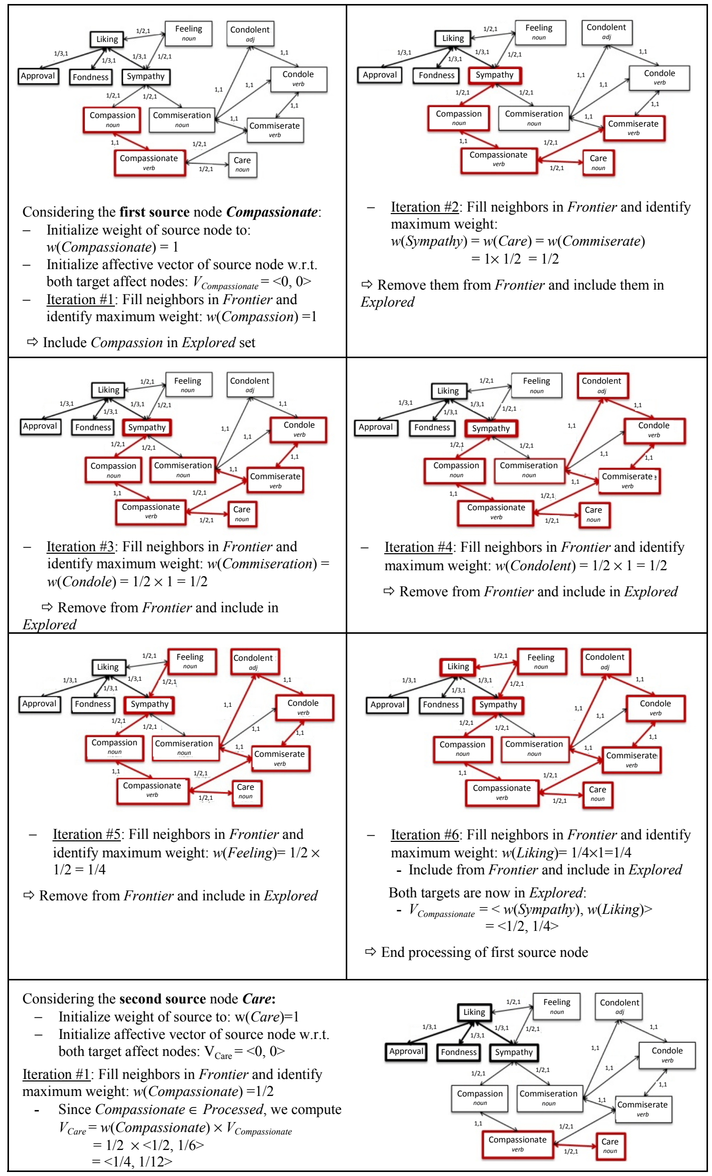

Fig. 6. Sample run of *Max_Affect*, from source words: *Compassionate* and *Care*, to destination affects: *Sympathy* and *Liking*

Consider the sample LAG in Fig. 5, where edge weights are computed following their semantic relationship reliability using Formula 3. Fig. 6 shows the result of a sample run of *Max_Affect*, considering as source: word concept nodes *Compassionate* and *Care*, and as destination: affect nodes *Sympathy* and *Liking*.

3. Problems with Max_Affect

While *Max_Affect* provides a solution to perform LSA in a completely unsupervised manner, nonetheless, it suffers from two main drawbacks regarding: i) effectiveness and ii) efficiency.

In terms of effectiveness, we realized that semantic connectivity between affect concepts in the LAG does not always accurately portray their affective expressiveness. For instance, considering the LAG extract of Fig. 5, we can reach affect node *liking* from affect node *sympathy* through concept node *feeling* with a higher weight compared with the direct link between *sympathy* and *liking*, i.e., *w*(*feeling*, *sympathy*)=1/2> and *w*(*liking*, *sympathy*)=1/3. In the example in Fig. 6, this led to $V_{Compassion} = <1/2, 1/4>$ and $V_{Care} = <1/4,$

1/8> (let us refer to this as result #1). Had we disregarded concept node *feeling* which connects *liking* with *sympathy*, and only used the direct affective connection between the latter two, we would have obtained $V_{Compassion}$ = <1/2, 1/6> and V_{Care} = <1/4, 1/12> (let's refer to this as result #2). At first glance, both results sound reasonable, and one cannot really judge which is better and which is worse. Yet, after empirically testing *Max_Affect* on the manually annotated ANEW word dataset [5, 63] (cf. Section IV) and investigating *Max_Affect*'s produced scores, we realized that connections between affect concepts from WNAH are more reliable in carrying sentiment expressiveness compared with lexical and semantic connections from WordNet. Yet, the logic tends to break down when propagating weight scores between the affect nodes. For instance, crossing from *sympathy* to *liking* should carry the whole weight of *sympathy* toward *liking*, *w*(*liking*, *sympathy*)=1, and not the other way around, since *sympathy*-*IsA*-*liking* where *IsA* (*hyponymy*) is a reliable (sentiment conductive) relationship. In other words, reaching affect node *sympathy* from any concept node c_i should be enough to identify c_i's sentiment weight w.r.t. affect *liking*, i.e., $w_{liking}(c_i) = w_{sympathy}(c_i)$ (e.g., considering concept node *compassionate* in Fig. 5, we would expect $w_{liking}(compassionate) = w_{sympathy}(compassionate) = 1/2\times1\times1/2\times1=1/4$).

As for efficiency, *Max_Affect* requires average polynomial (quadratic) time w.r.t. the size of the LAG covered in the navigation process (from source concept nodes to target affect nodes) which, despite LAG navigation optimizations and parallelization, remained relatively time consuming. This led us to provide an improved solution, considering the above mentioned effectiveness and efficiency issues in designing LISA 2.0.

C. LISA 2.0 – Affect Propagation and Lookup

To address the issues mentioned above, LISA 2.0 includes three main components: i) *WNAH_Propagation* to handle affect score computation between affect nodes themselves considering their affective connections only, while disregarding word concepts and their lexical/semantic connections in the LAG (this allowed solving the LAG lexical/semantic connectivity problem of LISA 1.0), ii) *Back_Propagation* which propagates affect scores from user chosen affect nodes to all connected concept nodes in the LAG[2]. The set of affect-scored concepts form a *sentiment lexicon* which can be efficiently searched by iii) *Affect_Lookup* to identify word concept affect scores (handling LISA 1.0's efficiency problem). We describe LISA 2.0's components in following sub-sections.

1. WNAH_Propagation component

This component computes the sentiment scores of every affect node w.r.t. every other affect node in WNAH, such that each affect category becomes fully representative of all of the others. In other words, every affect node a_j in WNAH will be associated with a sentiment vector V_j consisting of 294 dimensions, where every dimension represents every other affect node in WNAH with its corresponding affect score w.r.t. a_j. On the one hand, this allows disregarding all lexical and sentiment concepts and connections when navigating between affect nodes in the LAG. On the other hand, instead of computing the maximum weight path between a word concept node c_i and all 294 affect nodes to get their sentiment scores (following *Max_Affect*, cf. Section III.B), we only need to compute the path from c_i to the closest affect node a_j, where a_j would provide through its sentiment vector V_j all the scores for all other WNAH affect categories. The sentiment vector of c_i, V_i would be equal to V_j multiplied by the maximum path weight from c_i to a_j, i.e., $V_j = w(c_i, a_j)\times V_i$. Note that Affect nodes are processed in parallel, where affect vectors are computed independently in every iteration. Then, we iterate once for every inner node in the hierarchy, processing all vectors in parallel in order to update their weights w.r.t. inner node connectivity. For instance, a node a_i having node a_j as its parent (or child), will have its affect vector updated w.r.t. the latter's, by multiplying their weights while preserving the maximum weight following every vector dimension. Consider for instance the sample affective hierarchy in Fig. 4 extracted from WNAH, where edge weights are computed following *hypernymy/hyponymy* affective reliability (conductance) following Formula 3. Fig. 7 shows the affect vectors resulting from the execution of *WNAH_Propagation* on the LAG in Fig. 5 w.r.t. its affect hierarchy extracted from Fig. 7.

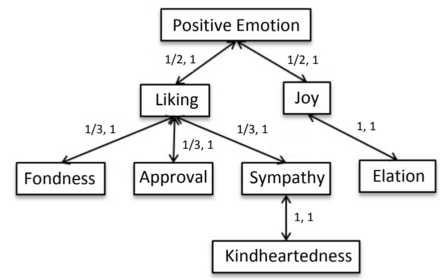

Fig. 7. Extract of the WNAH hierarchy

$$\begin{array}{cc}
 & \begin{array}{cccccccc} PosEm & Liking & Joy & Fond & Appr & Symp & Elation & Kind \end{array} \\
\begin{array}{c} V_{PosEm} \\ V_{Liking} \\ V_{Joy} \\ V_{Fond} \\ V_{Appr} \\ V_{Symp} \\ V_{Elation} \\ V_{Kind} \end{array} &
\left[\begin{array}{cccccccc}
1 & 1/2 & 1/2 & 1/6 & 1/6 & 1/6 & 1/2 & 1/6 \\
1 & 1 & 1/2 & 1/3 & 1/3 & 1/3 & 1/2 & 1/3 \\
1 & 1/2 & 1 & 1/6 & 1/6 & 1/6 & 1 & 1/6 \\
1 & 1 & 1/2 & 1 & 1/3 & 1/3 & 1/2 & 1/3 \\
1 & 1 & 1/2 & 1/3 & 1 & 1/3 & 1/2 & 1/3 \\
1 & 1 & 1/2 & 1/3 & 1/3 & 1 & 1/2 & 1 \\
1 & 1/2 & 1 & 1/6 & 1/6 & 1/6 & 1 & 1/6 \\
1 & 1 & 1/2 & 1/3 & 1/3 & 1 & 1/2 & 1
\end{array}\right]
\end{array}$$

Fig. 8. Affect vectors for every lexical node in the LAG of Fig 5

After computing all vectors for all affect nodes in WNAH, every affect node becomes fully descriptive of the affective scores of all other nodes in WNAH, such that accessing any affect node would give away all of WNAH's sentiment descriptiveness.

2. WNAH_Propagation component

Having computed the affect scores of all affect nodes in WNAH (using *WNA_Propagation*), the *Back_Propagation* component propagates the produced affect scores from user chosen affect nodes to all connected concept nodes in the LAG. As a result, all lexical concepts connected with any affect node acquires an affect score, form a *sentiment lexicon*. The latter can then be utilized to perform sentiment analysis by looking-up the affect vectors of the target lexical concepts from the lexicon.

The *Back_Propagation* component is a variation of *Dijkstra*'s maximum weight process utilized in *Max_Affect*, with the following modifications: i) a set of source affect nodes $A \in$ G along with their affect vectors ∇_A (pre-computed using *WNA_Propagation*); it does not require a set of lexical concept nodes as input since it will process all of them $\in G$, ii) it navigates the LAG starting from all source affect nodes in parallel (with a dedicated thread assigned to every

[2] Recall that our approach is different from existing graph-based LSA methods in that we distinguish the affect concept hierarchy from the lexical KB, to process affective concepts separately following their affective relationships, before mapping them with their lexical counterparts with their lexical and semantic connections. To do so, we consider affective and lexical/semantic relationships and their weight combinations differently as discussed in Section III.B and C.

source node), where affect vectors are computed independently in every iteration, iii) it navigates from every source affect node toward its surrounding concept nodes and beyond, back-propagating toward all connected concept nodes, iv) affect vectors of lexical nodes are computed directly from those of their connected affect node vectors, and v) the maximum affect vector weights for all concept nodes produced from every source affect node (i.e., from every thread) are finally retained. The pseudo-code for *Back_Propagation* is provided in [18]. Consider the same sample LAG example in Fig. 5, Fig. 9 shows the result of a sample run of *Back_Propagation*, starting from the affective nodes in the LAG and propagating their affective scores (in parallel) toward all lexical concept nodes in the graph.

$$\begin{array}{c} \\ V_{Liking} \\ V_{Fond} \\ V_{Symp} \\ V_{Appr} \end{array} \begin{array}{c} \begin{array}{cccc} \textit{Liking} & \textit{Fond} & \textit{Symp} & \textit{Appr} \end{array} \\ \begin{bmatrix} 1 & 1/3 & 1/3 & 1/3 \\ 1 & 1 & 1/3 & 1/3 \\ 1 & 1/3 & 1 & 1/3 \\ 1 & 1/3 & 1/3 & 1 \end{bmatrix} \end{array}$$

a. Input affect vectors for affect nodes in the LAG from Fig. 5

$$\begin{array}{c} \\ V_{Feeling} \\ V_{Compassion} \\ V_{Commiseration} \\ V_{Compassionate} \\ V_{Condolent} \\ V_{Condole} \\ V_{Commiserate} \\ V_{Care} \end{array} \begin{array}{c} \begin{array}{cccc} \textit{Liking} & \textit{Fond} & \textit{Symp} & \textit{Appr} \end{array} \\ \begin{bmatrix} 1 & 1/3 & 1 & 1/3 \\ 1/2 & 1/6 & 1/2 & 1/6 \\ 1/2 & 1/6 & 1/2 & 1/6 \\ 1/2 & 1/6 & 1/2 & 1/6 \\ 1/2 & 1/6 & 1/2 & 1/6 \\ 1/2 & 1/6 & 1/2 & 1/6 \\ 1/2 & 1/6 & 1/2 & 1/6 \\ 1/4 & 1/12 & 1/4 & 1/12 \end{bmatrix} \end{array}$$

b. Final affective weights

Fig. 9. Input and result of a sample run of *WNAH_Propagation*

3. *Affect_Lookup component*

The resulting set of affect-scored concepts forms a *sentiment lexicon* which can be efficiently searched to lookup any word concept affect score. For instance, the affect score of concept *care* w.r.t. affect category *approval* can be directly identified as =1/12 by looking it up from ∇_C. This is handled by the *Affect_Lookup* component, which makes use of legacy indexing techniques (e.g., B+ Tree [16]) to access and efficiently search ∇_C. We do not describe *Affect_Lookup* further here since it comes down to a typical data lookup process.

To sum-up, the LISA 2.0 module, through its *Affect_Lookup* component (which makes use of the sentiment lexicon produced by *Back-Popagation* and *WNA_Propagation*), allows to transform the problem of LSA from a (polynomial) graph navigation problem (with LISA 1.0) into a fast (logarithmic) data (lexicon) lookup problem. At the same time, LISA 2.0's lexicon construction process (through *Back-Popagation* and *WNA_Propagation*) is fully automated and does not require any training or manual effort.

IV. Experimental Evaluation

We first describe the experimental data and pre-processing in Section IV.A, before presenting and discussing polarity and affect evaluation results in Sections IV.B and IV.C respectively. In summary, results show that LISA 2.0 outperforms LISA 1.0 in both LSA quality and performance, while being on a par with existing supervised approaches (without the need for training or manual effort).

A. *Experiemtal Data*

We utilized the ANEW (Affective Norms for English Words) dataset [5, 63] to evaluate LISA 1.0 and 2.0. ANEW consists of 1024 words in the English language, manually rated in terms of *pleasure*, *arousal*, *dominance* in [5] as well as *happiness*, *anger*, *sadness*, *fear*, and *dislike/disgust* in [63]. Ratings were conducted by a large number of psychology students equally distributed between female and male candidates. Ratings for every dimension were provided on a 9-point scale in [5] and on a 5-point scale in [63], which can be translated into integers ($\in [1, 9]$ or $\in [1, 5]$) designating [*min*, *max*] expressiveness. For instance, *pleasure* was rated from *no pleasure* (=1) to *extreme pleasure* (=9), and arousal from *not aroused* (=1) to *extremely aroused* (=9). Here, we normalized ANEW's ratings to obtain scores $\in [0, 1]$, representing them in a common referential which would be easier to compare with LISA and other existing LSA methods. As for the afore mentioned dimensions, we considered *pleasure* to describe word polarity (*negative*-to-*positive*), and *happiness*, *anger*, *sadness*, *fear*, and *dislike/disgust* to describe their respective affect categories[3]. Note that certain existing LSA methods, e.g., [4, 6, 32], produce polarity scores $\in [-1, 1]$, varying from absolutely *negative* (score=-1) to absolutely *positive* (score=1). The latter were also normalized to the [0, 1]. As for LISA, sentiment scores are inherently $\in [0, 1]$ following the weight cost model and navigation processes adopted in our approach.

B. *Polarity Evaluation*

We compared LISA 1.0 and 2.0 with ANEW and two recent polarity detection methods available online: SentiWordNet [4], and AlchemyAPI [32]. The results of alternative solutions were produced based on the sentiment scores extracted from their original studies (available online). For the two latter methods, we identified the ANEW words matching with the corresponding lexicon entries to produce the corresponding polarity scores. A snapshot of the results is provided in Fig. 10. The complete set of empirical graphs and data is provided online[4]. Results are summarized Table 1. Fig. 10 shows *positive* polarity scores w.r.t. ANEW, where words have been ranked following ANEW's *positive* intensities (from highest to lowest). Similar graphs were produced for *negative* polarity scores and are provided online. Three main observations can be made. First, one can realize that LISA 2.0 produced results which are more consistently distributed following ANEW's ratings compared with LISA 1.0. Second, LISA 2.0's results show concentrations of score points around the ANEW reference score line, with clusters of points forming around *positive* polarity scores = 0.8, 0.64, 0.5, 0.37, and 0.18 (highlighted in Fig. 10.b) following ANEW's slope. This highlights LISA 2.0's quality in producing scores which correlate more closely with ANEW's manual ratings compared with LISA 1.0. Third, most alternative solutions which are supervised produce polarity scores which are relatively dispersed in the polarity space (cf. Fig. 10.c and d). This reflects their supervised learning nature, which produces results that are varied and reflective of the diversity of their training data, compared with LISA's less dispersed and more rigorously structured (clustered) results, reflecting the structured nature of its LAG reference and graph computation process.

Pearson Correlation Coefficient (PCC) results compiled in Table 1 show that LISA 2.0's performance is on a par with existing (supervised learning) approaches. IBM's AlchemyAPI opinion mining engine produced the best results, distinctively surpassing the other approaches including LISA.

C. *Affect Evaluation*

We also compared LISA 1.0 and 2.0 with ANEW as well as two alternative affect analysis methods available online: EmoSenticNet [56], and Tone Analyzer [33]. A snapshot of the results is provided in Fig. 11. The complete set of empirical graphs and corresponding data is provided online[5]. Results are summarized Table 2.

Fig. 11 shows *dislike/disgust* polarity scores w.r.t. ANEW, where words have been ranked following ANEW's *dislike* intensities (from highest to lowest). Similar graphs were produced for the other four affective categories (i.e., *happiness*, *anger*, *sadness*, *fear*) and

[3] We disregard *arousal* and *dominance* in our current experiments since they reflect *behavioral* rather than *affective* dimensions [10].

[4] http://sigappfr.acm.org/Projects/LISA

are provided online. Results here reflect observations similar to the ones made earlier with polarity scores: i) LISA 2.0 produced results which are more evenly distributed along ANEW's ratings compared with LISA 1.0, ii) LISA 2.0's results show concentrations of score points around the ANEW score line, with clusters of points forming around *dislike* intensity scores = 0.67, 0.46, 0.32, 0.18, and 0.09 (highlighted in Fig. 11.b), iii) IBM's Tone Analyzer, which is a supervised learning solution, produced affect scores that are relatively dispersed in the affective space (cf. Fig. 11.d), compared with LISA's clustered results, reflecting the former's supervised learning nature and the diversity of the training data, iv) EmoSenticNet, which is a semi-supervised sentiment lexicon, produced discrete affect category labels (in the form of scores ∈ {0, 1}, where the score of a word that belongs to the category =1, otherwise, it is =0). It does not produce affective intensity levels as clearly reflected in the binary nature of its results (in Fig. 11.c).

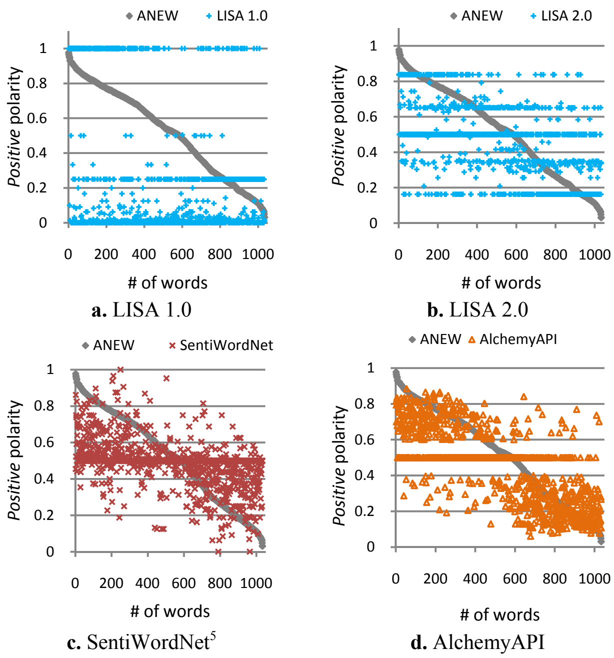

Fig. 10. *Positive* polarity scores w.r.t. the ANEW dataset

Table 1. Average *PCC* scores for *positive* and *negative* polarity

	Positive	*Negative*	*Avg.*
AlchemyAPI	0.7476	0.7477	0.7477
SentiWordNet 3	0.4934	0.4934	0.4934
LISA 2.0	0.4496	0.4496	0.4496
SenticNet 3	0.4504	0.4343	0.4424
LISA 1.0	0.2497	0.1872	0.2185

PCC results compiled in Table 2 show that LISA 2.0's performance, is on a par with existing supervised approaches. IBM's Tone Analyzer results, while more varied and dispersed than LISA's, slightly surpassed the latter's effectiveness w.r.t. the ANEW experimental dataset. This highlights LISA's potential as an unsupervised word-level LSA method capable of contending with existing supervised solutions. Yet, we clarify that LISA only performs word-level analysis at this stage, while Tone Analyzer is capable of sentence and document-level analyses.

The reader can refer to [18] for a more detailed description of the experimental results, as well as the whole framework.

[5] SenticNet results are close to those of SentiWordNet and are omitted here.

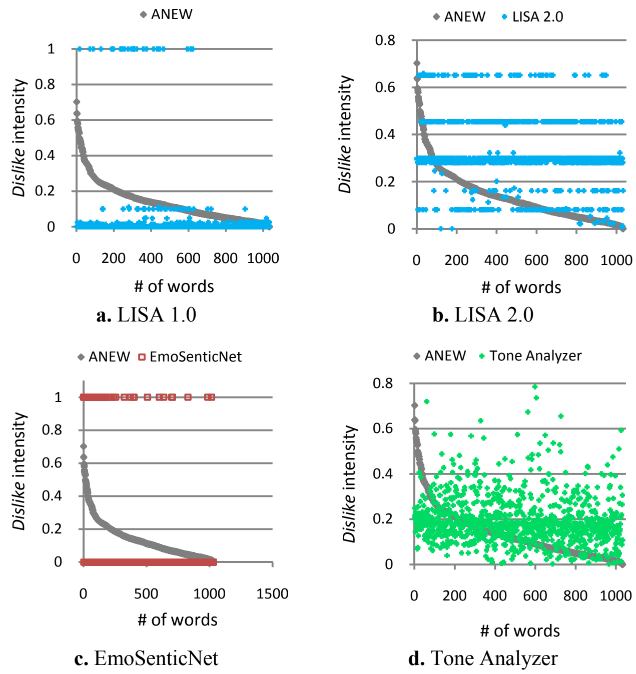

Fig. 11. *Dislike/disgust* affective scores w.r.t. the ANEW dataset

Table 2. Average *PCC* scores for *happiness*, *anger*, *sadness*, *fear*, and *dislike/disgust* affective categories

	Happiness	*Anger*	*Sadness*	*Fear*	*Dislike*	*Avg.*
Tone Analyzer	0.1997	0.1488	0.1299	0.0756	0.1513	0.14106
LISA 2.0	0.2251	0.1667	0.0108	0.0807	0.1669	0.13004
EmoSenticNet	0.1512	-0.0369	0.0394	0.0838	0.0671	0.06092
LISA 1.0	0.1257	0.0697	0.0045	0.0338	0.0698	0.0607

V. CONCLUSION

This paper introduces LISA, an unsupervised word-level knowledge graph-based LSA solution, which uses different variants of shortest path graph navigation techniques to compute and propagate affective scores in a lexical-affective graph (LAG). LISA was designed in two iterations, producing two modules: i) LISA 1.0 for affect navigation, and ii) LISA 2.0 for affect propagation and lookup. LISA 1.0 suffered from the semantic connectivity problem shared by some existing lexicon-based methods, and required polynomial execution time. This led to the development of LISA 2.0, which i) processes affective relationships separately from lexical/semantic connections (solving the semantic connectivity problem of LISA 1.0), and ii) produces a *sentiment lexicon* which can be searched in logarithmic time (handling LISA 1.0's efficiency problem). Experiments on the ANEW dataset show that LISA 2.0 outperforms LISA 1.0 in both LSA quality and performance, while being on a par with existing supervised solutions (without the need for training or manual effort).

We are currently investigating phrase-level and sentence-level LSA, combining LISA's functionality with context-level features such as word associations, valence shifters, and a dedicated emoji affect lexicon [22], to perform unsupervised LSA on short social media texts. In the near future, we aim to explore *implicit semantics* (a.k.a. *latent semantics*) [68] which can be inferred from the statistical analysis of word/phrase embeddings (feature vectors), following their co-occurrence in a corpus [29] (e.g., identifying that "failure" is *related to* "sadness" following their feature vector offsets). We aim to investigate the latter considering our LAG structure, combined with graph-based indexing approaches, e.g., [70, 71], toward unsupervised knowledge-based and corpus-based LSA.

ACKNOWLEDGMENTS

This study is funded by the National Council for Scientific Research Lebanon (CNRS-L), the Lebanese American Univ. (LAU), and the Research Support Foundation of the State of Sao Paulo (FAPESP).

REFERENCES

[1] Abbasi A., et al., *Affect Analysis of Web Forums and Blogs Using Correlation Ensembles.* IEEE Tran. on Knowledge & Data Engineering (TKDE), 2008. 20(9):1168-1180.

[2] Akhtar M.S., et al., *Feature Selection and Ensemble Construction: A Two-step Method for Aspect based Sentiment Analysis.* Knowledge-based Systems, 2017. 125: 116-135.

[3] Appel O., et al., *A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level.* Knowledge-Based Systems, 2016. 108: 110-124.

[4] Baccianella S., et al., *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.* Language Resources & Eval. (LREC), 2010, 2200-2004.

[5] Bradley M. and Lang P., *Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings.* Technical Report C-1, 1999. University of Florida.

[6] Cambria E., et al., *SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis.* Inter. AAAI Conf. 2014. pp. 1515-1521.

[7] Cambria E., et al., *SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives.* Inter. Conf. on Comput. Linguistics (COLING) 2016, 2666-2677.

[8] Cambria E., et al., *SenticNet 5: Discovering Conceptual Primitives for Sentiment Analysis by Means of Context Embeddings.* Inter. AAAI Conf., 2018, 1795-1802.

[9] Charbel N., et al., *Resolving XML Semantic Ambiguity.* International Conference on Extending Database Technology (EDBT'15), 2015. Brussels, Belgium, pp 277-288.

[10] Chuang Z. and Wu C., *Multi-Modal Emotion Recognition from Speech and Text.* Computational Linguistics and Chinese Language Processing, 2004. 9(2): 45-62.

[11] Cormen T.H., et al., *Introduction to Algorithms (3rd Ed.).* MIT Press, 2009.

[12] da Silva N., et al., *A Survey and Comparative Study of Tweet Sentiment Analysis via Semi-Supervised Learning.* ACM Comput. Surv. , 2016. 49(1): 15:1-15:26.

[13] de Albornoz J., et al., *SentiSense: An easily scalable concept-based affective lexicon for sentiment analysis.* Language Resources & Evaluation (LREC) 2012, 3562-3567.

[14] Donath J., et al., *Visualizing Conversation.* Computer-Mediated Comm., 1999. 4(4).

[15] Ebrahimi M., et al., *Challenges of Sentiment Analysis for Dynamic Events.* IEEE Intelligent Systems, 2017. 32(5):70-75.

[16] Elmasri R. and Navathe S.B., *Fundamentals of Database Systems (6th ed.).* Upper Saddle River, N.J.: Pearson Education, 2010. pp. 652–660.

[17] Esparza G.G. et al., *A Sentiment Analysis Model to Analyze Students Reviews of Teacher Performance Using Support Vector Machines.* Inter. DCAI Conf. 2017, pp. 157-164.

[18] Fares M., et al., *Unsupervised Word-level Affect Analysis and Propagation in a Lexical Knowledge Graph.* Slesevier Knowledge-Based Systems, 2019. 165: 432-459.

[19] Ferilli S., et al., *Towards Sentiment and Emotion Analysis of User Feedback for Digital Libraries.* Italian Research Conf. on Digital Libraries (IRCDL'16), 2016. pp. 137-149.

[20] Francisco V., et al., *Ontological Reasoning for Improving the Treatment of Emotions in Text.* Knowl. Inf. Syst. , 2010. 25(3): 421-443.

[21] Fu X., et al., *Neurocomputing* 241: 18-27, Combine HowNet Lexicon to Train Phrase Recursive Autoencoder for Sentence-Level Sentiment Analysis.

[22] Gavilanes M.F. et al., *Creating Emoji Lexica from Unsupervised Sentiment Analysis of their Descriptions.* Expert Syst. Appl., 2018. 103: 74-91.

[23] Ghiassi M. and Lee S., *A Domain Transferable Lexicon Set for Twitter Sentiment Analysis using a Supervised Machine Learning Approach.* Expert Syst. Appl. , 2018. 106: 197-216.

[24] Giachanou A. and Crestani F., *Like It or Not: A Survey of Twitter Sentiment Analysis Methods.* ACM Comput. Surv., 2016. 49(2): 28:1-28:41

[25] Gill A.J., et al., *Identifying Emotional Characteristics from Short Blog Texts.* 30th Annual Meeting of the Cognitive Science Society, 2008. pp. 2237-2242.

[26] Godbole N., et al., *Large-Scale Sentiment Analysis for News and Blogs.* International Conference on Weblogs and Social Media (ICWSM'07), 2007. p. 4.

[27] Hatzivassiloglou V. and McKeown K.R., *Predicting the Semantic Orientation of Adjectives.* Meeting of the Asso. for Comput. Linguistics (ACL'97), 1997. pp. 174-181.

[28] Hearst M.A., *Direction-Based Text Interpretation as an Information Access Refinement.* Text-Based Intell. Sys.: Current Research & Practice in Info. Extract. & Retrieval 1992.

[29] Hill F., et al., *Learning Distributed Representations of Sentences from Unlabelled Data.* Conf. of the North American Ch. of the Asso. for Comput. Linguistics (ACL), 1367-1377.

[30] Hoffart J., et al., *YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia.* Artif. Intell., 2013. 194: 28-61.

[31] Hovy, E., *What are Sentiment, Affect, and Emotion? Applying the Methodology of Michael Zock to Sentiment Analysis.* Lang. Production, Cognition, & the Lexicon, 48, 2015, 13-24.

[32] IBM, *AlchemyAPI.* 2005. https://www.ibm.com/watson/alchemy-api.html, May 2019.

[33] IBM, *Tone Analyzer.* https://www.ibm.com/watson/services/tone-analyzer/, May 2019.

[34] Jaggi M., et al., *Swiss-Chocolate: Sentiment Detection using Sparse SVMs and Part-Of-Speech n-Grams.* Inter. Conf. on Comput. Linguistics (COLING), 2014, 601-604.

[35] Jiménez F.V., et al., *Simple Window Selection Strategies for the Simplified Lesk Algorithm for Word Sense Disambiguation.* Inter MICAI Conf., 2013. pp. 217-227.

[36] Kamps J., et al., *Using WordNet to Measure Semantic Orientations of Adjectives.* Language Resources and Evaluation (LREC'04), 2004. p. 4.

[37] Kang H., et al., *Senti-Lexicon and Improved Naïve Bayes Algorithms for Sentiment Analysis of Restaurant Reviews* Expert Syst. Appl., 2012. 39:6000–6010.

[38] Khan E.F., et al., *A Semi-Supervised Approach to Sentiment Analysis using Revised Sentiment Strength based on SentiWordNet.* Knowl. Inf. Syst. , 2017. 51(3): 851-872.

[39] Khan F.H., et al., *Lexicon-based Semantic Detection of Sentiments using Expected Likelihood Estimate Smoothed Odds Ratio.* Artif. Intell. Rev., 2017. 48(1): 113-138

[40] Kim S. and Hovy E., *Determining the Sentiment of Opinions.* Inter. Conf. Computational Linguistics (COLING ’04), 2004. pp. 1367-1373.

[41] Koc S.S., et al., *Triadic Co-Clustering of Users, Issues and Sentiments in Political Tweets.* Expert Syst. Appl., 2018. 100: 79-94.

[42] Korayem M., et al., *Sentiment/subjectivity analysis survey for languages other than English.* Social Netw. Analys. Mining, 2016. 6(1): 75:1-75:17.

[43] Lee G. et al., *Sentiment Classification with Word Localization based on Weakly Supervised Learning with a Convolutional Neural Network* Knowl.-Based Syst. , 2018. 152: 70-82.

[44] Li S., et al., *Exploiting Co-occurrence Opinion Words for Semi-supervised Sentiment Classification.* Advanced Data Mining and Applications (ADMA'13), 2013. pp. 36-47.

[45] Liu H., et al., *A Model of Textual Affect Sensing Using Real-World Knowledge.* Proc. Eighth Int’l Conf. Intelligent User Interfaces (IUI'03), 2003. pp. 125-132.

[46] Ma C., et al., *Topic and Sentiment Unification Maximum Entropy Model for Online Review Analysis.* International World Wide Web Conferences (WWW'15), 2015. pp. 649-654.

[47] Martín-Wanton T., et al., *Opinion Polarity Detection - Using Word Sense Disambiguation to Determine the Polarity of Opinions.* Inter. ICAART Conf., 2010. 1: 483-486.

[48] Miller G.A. and Fellbaum C., *WordNet Then and Now.* Language Resources and Evaluation (LRE), 2007. 41(2): 209-214.

[49] Mishne G., *Experiments with Mood Classification.* Proc. First Workshop on Stylistic Analysis of Text for Information Access Workshop (Style), 2005. p. 8.

[50] Mohammad S.M. and Turney P.D., *Emotions Evoked by CommonWords and Phrases: Using Mechanical Turk to Create an Emotion Lexicon.* NAACL HLT, 2010, 26–34.

[51] Mondal A., et al., *Employing sentiment-based affinity and gravity scores to identify relations of medical concepts.* IEEE Symp. on Comput. Intelligence (SSCI), 2017. pp. 1-7.

[52] Mullen T. and Collier N., *Sentiment Analysis using Support Vector Machines with Diverse Information Sources.* Empirical Methods in NLP (EMNLP'04), 2004. pp. 412–418.

[53] Neviarouskaya A., et al., *Textual Affect Sensing for Sociable and Expressive Online Communication.* Affective Computing & Intelligent Interaction (ACII'07) 2007, 218-229.

[54] Parrot, W.G., *Emotions in Social Psychology: Key Readings (Key Readings in Social Psychology).* Psychology Press, 1st Edition, 2001. p. 392.

[55] Pennebaker J., et al., *Linguistic Inquiry and Word Count: LIWC.* LIWC Manual, 2007, 22.

[56] Poria S., et al., *Enhanced SenticNet with Affective Labels for Concept-based Opinion Mining.* IEEE Intelligent Systems, 2013. 28(2): 31–38.

[57] Poria S., et al., *EmoSenticSpace: A Novel Framework for Affective Common-Sense Reasoning.* Knowl.-Based Syst. , 2014. 69: 108-123.

[58] Rao Y., et al., *Intensive Maximum Entropy Model for Sentiment Classification of Short Text.* Database Systems for Advanced Applications (DASFAA) Workshops, 2015, 42-51.

[59] Ravi K. and Ravi V., *A survey on opinion mining and sentiment analysis: Tasks, approaches and applications.* Knowledge-Based Systems, 2015. 89:14–46.

[60] Scherer K.R., *What are Emotions? And how can they be Measured?* Social Science Information, 2005. 44(4):693–727.

[61] Schouten K., et al., *Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis with Co-occurrence Data.* IEEE Trans. Cybernetics, 2018. 48(4): 1263-1275.

[62] Singh P., et al., *Commonsense: Knowledge Acquisition from the General Public.* Inter. Conf. on Onto., DBs, & Apps. of Semantics for Large Scale IS (ODBASE), 2002. p. 7.

[63] Stevenson R.A., et al., *Manual for the Categorization of the Affective Norms for English Words (ANEW) by Discrete Emotions.* Indiana University Bloomington, 2014. p. 123.

[64] Stone P.J. and Hunt E.B. , *A Computer Qpproach to Content Qnalysis: Studies using the General Inquirer System.* Spring Joint Computer Conf. (AFIPS'63), 1963. pp. 241–256.

[65] Strapparava C., et al., *The Affective Weight of Lexicon.* Language Resources and Evaluation (LREC'06), 2006. pp. 423-426.

[66] Subasic P. and Huettner A., *Affect Analysis of Text Using Fuzzy Semantic Typing.* IEEE Trans. Fuzzy Systems, 2001. 9(4):483-496.

[67] Taddesse F.G., et al., *Semantic-based Merging of RSS Items.* World Wide Web Journal, 2010, 13(1-2): 169-207, Springer Netherlands.

[68] Tekli J., *An Overview on XML Semantic Disambiguation from Unstructured Text to Semi-Structured Data: Background, Applications, & Ongoing Challenges.* IEEE Trans. on Knowledge and Data Engineering (TKDE), 2016. 28(6): 1383-1407.

[69] Tekli J., et al., *Building Semantic Trees from XML Documents.* Journal of Web Semantics (JWS), 2016. 37–38:1–24.

[70] Tekli J., et al., *SemIndex+: A Semantic Indexing Scheme for Structured, Unstructured, and Partly Structured Data.* Knowledge-Based Systems, 2019. 164: 378-403.

[71] Tekli J., et al., *Full-fledged Semantic Indexing and Querying Model Designed for Seamless Integration in Legacy RDBMS.* Data and Knowledge Engineering, 2018. 117: 133-173.

[72] Trainor K.J. et al., *Social Media Technology Usage and Customer Relationship Performance: a Capabilities-based Examination of Social CRM.* Journal of Business Research, 2013. 67(6): 1201-1208.

[73] Tsirakis N., et al., *Large Scale Opinion Mining for Social News and Blog Data.* Journal of Systems and Software, 2017. 127: 237-248.

[74] Turney P.D. and Littman M.L., *Measuring Praise and Criticism: Inference of Semantic Orientation from Association.* ACM Trans. Information Systems, 2003. 21(4):315-346.

[75] Valdivia A., et al., *Sentiment Analysis in TripAdvisor.* IEEE Intell. Sys., 2017. 32:72-77.

[76] Valitutti A., et al., *Developing Affective Lexical Resources.* PsychNology, 2004. 2:61-83.

[77] Vasilescu F., et al., *Evaluating Variants of the Lesk Approach for Disambiguating Words.* Language Resources and Evaluation (LREC'04), 2004. pp. 633-636.

[78] Vilares D., et al., *Universal, Unsupervised (Rule-based), Uncovered Sentiment Analysis.* Knowl.-Based Syst., 2017. 118: 45-55.

[79] Wang G., et al., *POS-RS: A Random Subspace Method for Sentiment Classification based on Part-Of-Speech Analysis.* Inf. Process. Manage. , 2015. 51(4): 458-479.

[80] Wawer A. and Rogozinska D., *How Much Supervision? Corpus-Based Lexeme Sentiment Estimation.* Inter. Conf. on Data Mining (ICDM'12) Workshops, 2012. pp. 724-730.

[81] Weichselbraun A., et al., *Enriching Semantic Knowledge Bases for Opinion Mining in Big Data Applications.* Knowl.-Based Syst., 2014. 69: 78-85.

[82] Wilson T., et al., *Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis.* Computational Linguistics, 2009. 35(3): 399-433.

[83] Yeh J.F., et al., *Dimensional Sentiment Analysis in Valence-Arousal for Chinese Words by Linear Regression.* Inter. Conf. on Asian Language Processing (IALP'16) 2016, 328-331.

[84] Zhang S., et al., *Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary.* Future Generation Comp. Syst., 2018. 81: 395-403.

[85] Zhang Y., *Incorporating Phrase-level Sentiment Analysis on Textual Reviews for Personalized Recommendation.* Inter. WSDM Conf., 2015. pp. 435-440.

[86] Zhang Y., et al., *Unsupervised Sentiment Analysis of Twitter Posts Using Density Matrix Representation.* European Conf. on Information Retrieval (ECIR'18), 2018. pp. 316-329.

[87] Zhao Y., et al., *Appraisal Expression Recognition with Syntactic Path for Sentence Sentiment Classification.* Int. J. Comput. Proc. Oriental Lang., 2011. 23(1): 21-37.