# Quantifying the Natural Sentiment Strength of Polar Term Senses Using Semantic Gloss Information and Degree Adverbs

Mohammad Darwich[1], Shahrul Azman Mohd Noah[2], Nazlia Omar[2], Nurul Aida Osman[2], and Ibrahim Said Ahmad[3]
[1] Department of Computer Science, National University of Malaysia, Malaysia
[2] Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia
[3] Department of Information Technology, Bayero University Kano, Nigeria
Email: modarwish@hotmail.com, {shahrul, nazlia}@ukm.edu.my, n.aidaosman@gmail.com, isahmad.it@buk.edu.ng

***Abstract*—In Sentiment Analysis (SA), a vague assignment of a text to a set of *n-ary* discrete classes is insufficient. A great deal of research is concentrated on the automated assignment of strength to both terms and the finer-grained term senses, but these strength values rely purely on *statistical means*, and there is no *semantic mechanism* involved, leading to potentially biased results. As a solution, this works proposes a model that utilizes only the semantic information manually encoded within the human-defined glosses of term senses, a semantic network, and a set of predefined degree adverbs, in order to quantify their 'Natural' Sentiment Strength (NSS) values. The 'natural' sentiment strength of a term sense here refers to the strength value derived in a 'semantically natural' manner, i.e. the NSS is assigned based on the agreed-upon meanings that humans have naturally assigned to words; and not 'artificially statistical', i.e. based a simple metric of probabilistic computation. Intrinsic evaluation against a manually-annotated gold standard benchmark demonstrates that the model outperforms related sense-level lexicon generation models against this same benchmark, and that it is in agreement with human intuition.**

***Index Terms*—sentiment analysis, opinion mining, sentiment lexicon, sentiment strength, sentiment lexicon generation**

## I. Introduction

Sentiment Analysis (SA) is the detection of sentiment, attitude, opinion, and emotion in unstructured text. In SA, a sentiment expression is essentially a two-dimensional vector, with both a direction (polarity as positive or negative) and a magnitude (strength of the polarity). Therefore, a vague assignment of a text to a set of *n-ary* discrete classes is insufficient. Accurate SA models must be sensitive enough to detect the precise details in a sentiment expression, and then provide this information in numerical form, which is a more realistic representation of the sentiment strength value embedded within the target expression [1]-[3]. This finer-grained approach involves placing the sentiment of a text unit on a [+1, -1] continuum between positive and negative. This would allow for quantifying the precise degree a sentiment deviates from the norm, rather than rely on hard classification to a set of predefined discrete classes. Furthermore, this would be able to answer the question "Precisely how favorably or unfavorably does a person feel about a product, service, company, brand or political figure?"

Several works in the literature focus on assigning a full-text with a sentiment strength [4]-[7]. However, a recurring issue is that, in order to assign a text with a global sentiment strength, a list of individual terms tagged with a strength score is needed. Several works on manual compilation of sentiment strength dictionaries do exist [1], [7]-[9]. However, this task is onerous in terms of annotator time and effort [10], [11]. Hence, a great deal of research is concentrated on the automated assignment of strength values to terms [12]-[18]. While some prominent sense-level lexicon generation models avoid quantifying strength altogether [19], [20], the SentiWordNet 3.0 lexicon generation model [21] also uses statistical means (i.e., the agreement of a committee of eight supervised classifiers) as the quantification of a sense's sentiment strength.

However, this is not a valid representation of sentiment strength, since it tends to reflect the 'likeliness' of a term being in a particular class, or the confidence in the accuracy of the labeling process, rather than the 'natural' sentiment strength of the term. This is because the strength values rely purely on *statistical o probabilistic means*, and there is no *semantic mechanism* involved, leading to biased results. For instance, if a moderately polar term such as *good* has a higher occurrence frequency in the positive class than a strongly polar term such as *excellent*, the former would be assigned a higher strength than the latter. However, this should not be the case in reality. Upon inspection of the human-defined gloss information of these two terms, the latter should possess a higher strength than the former.

In this work, the 'Natural' Sentiment Strength (NSS) of a term sense refers to the strength value derived in a 'semantically natural' manner, i.e. the NSS is assigned based on the agreed-upon meanings that humans have

Manuscript received December 23, 2019; revised May 22, 2020.


doi: 10.12720/jait.11.3.109-118

naturally assigned to words; and not 'artificially statistical', i.e. based a simple metric of probabilistic computation, as in the majority of state-of-the-art sense-level sentiment lexicon generation models. In other words, in contrast to prior models that rely on statistical means to measure strength and in turn generate biased results, this model utilizes only the semantic information manually encoded within the human-defined glosses of term senses, a semantic network, and a set of predefined degree adverbs, in order to quantify their true NSS values. Intrinsic evaluation against a manually-annotated gold standard benchmark demonstrates that the model outperforms related models such as the popular SentiWordNet 3.0 lexicon generation model [21] against the same benchmark, and that it is in agreement with human intuition. This also demonstrates its practical application in real-world sentiment analysis tasks.

This paper is structured as follows. Section II presents related prior work. Section III presents the methodology carried out to develop the proposed model. Section IV presents the evaluation procedure, while Section V presents the results for this procedure. Section VI concludes.

## II. Prior Work

Sentiment Analysis (SA) is the detection of sentiment, attitude, opinion, and emotion in unstructured text. SA is itself a multi-faceted problem [22], [23]. According to [23], SA is among the factors that are important for the advancement of AI and its related fields. Other techniques are also important to consider such as the deep learning based aspect extraction for aspect-based SA [24], e.g., attentive Long Short-Term Memory (LSTM) [25]. The recent literature also considers techniques such as word representations and capsule networks, which are applied on SA in particular, but are also applicable on other challenging NLP applications [26]. Generally, there are two main approaches that are employed by SA models. The (unsupervised) lexicon-based approach involves making use of a sentiment lexicon to compute the overall sentiment polarity of a text document based on the aggregation of the polarity of the individual words embedded within the document [1]. The (supervised) classification-based approach involves constructing supervised machine learning classifiers that are fed with hand-labelled training data for the classification task [12].

SA applications can also highly benefit from sentiment strength, which refers to "the degree of intensity of the positivity or negativity of a sentiment expression" [4]. This allows SA applications to detect not only the polarity of the sentiment expressed, but also the strength of the polarity. Reference [13] demonstrates that the strength or intensity of polar (e.g. positive or negative) terms would allow for a finer-grained sentiment classification of text. Several works in the literature focus on assigning a full-text with a sentiment strength [1], [4]-[6], [12]. However, a recurring issue is that, in order to assign a text with a global sentiment strength, a list of individual terms tagged with a strength score is needed. Several works on manual compilation of sentiment strength dictionaries do exist [1], [7]-[9]. However, this task is even more onerous than the manual compilation of binary sentiment polarity dictionaries, since a numerical score must be generated as a representation of the strength.

Hence, a great deal of research is concentrated on the automated assignment of strength to terms. A work by [13] proposes a semi-supervised approach that utilizes sentiment-carrying word embeddings to generate a ranking for adjectives that have related semantic properties, achieving a corellation of 0.83 against a gold standard benchmark. Reference [14] develops a lexicon generation approach on the basis of constrained symmetric nonnegative matrix factorization, using both a dictionary and a social media corpus. Reference [15] proposes a random walk algorithm to generate a sentiment lexicon, where the time taken to reach a set of predefined seed words in the semantic network acts as an estimation of the polarity of a target word. Other works utilize semantic distance [16], semi-supervised label propagation [17], and bootstrapping from semantic relations [18]. However, a prominent issue is that all prior term-level lexicon generation models employ statistical or probabilistic information of a term's frequency in a certain class as representative of its sentiment strength, leading to potentially biased results. Moreover, these works focus on terms, and ignore the finer-grained term sense level.

While some prominent sense-level lexicon generation models avoid quantifying strength altogether [19], [20], the SentiWordNet 3.0 lexicon generation model [21] also uses statistical means (i.e., the agreement of a committee of eight supervised classifiers) as the quantification of a sense's sentiment strength. It can be argued that, semantically, this is an invalid representation of a sense's sentiment strength. This prominent limitation acts as the motivation to develop a model that utilizes only the semantic information manually encoded within the human-defined glosses of term senses, a semantic network, and a set of predefined degree adverbs, in order to quantify their true natural sentiment strength values.

## III. Quantifying the Natural Sentiment Strength of Polar Term Senses

This section presents all of the steps carried out to develop a model that quantifies the natural sentiment strength of polar term senses, in the generation of a sense-level sentiment lexicon. In other words, in contrast to prior sense-level models that rely on statistical means to measure strength and in turn generate biased results, this model utilizes purely semantic information.

The model is fully-unsupervised, since a polarity sentiment lexicon, the contextual gloss information and semantic network from a digital dictionary (e.g. WordNet), and a small set of manually-compiled degree adverbs, are the only source of input. Note that the polarity lexicon can be any a list of term senses tagged with a binary positive or negative polarity. There is no human involvement during execution of the model. The final output is a sense-level sentiment lexicon comprising subjective senses assigned

with their corresponding NSS scores. This final sentiment lexicon can aid in sense-level sentiment classification on full-text, and measure the precise NSS of the sentiment expressed within the text, as opposed to a hard classification of the text to predefined discrete classes (e.g., strongly positive, weakly positive, strongly negative, weakly negative). Objective term senses are not considered by this model, since they are sentiment-neutral, hence naturally carry an NSS of 0.

### A. Model Architecture

Fig. 1 presents a high-level architecture of the proposed model. This model comprises two main algorithms that work successively. The first is the Natural Sentiment Strength Quantification Algorithm (NSSQA), which uses contextual gloss information and a manually-compiled list of degree adverbs. The second is the Natural Sentiment Strength Quantification via Connectivity Algorithm (NSSQCA), which uses a semantic network to assist in the quantification of the remaining synsets in the lexicon that have not yet been labeled with an NSS. The input into this model is a set polar term senses, since the aim in this work is to focus on assigning polar term senses with a sentiment strength, with the assumption that they are already assigned with a polarity. This has implications in automatically labelling term- and sense-level polarity lexicons with NSS information. Finally, the model generates the NSS values for each of these polar term senses.

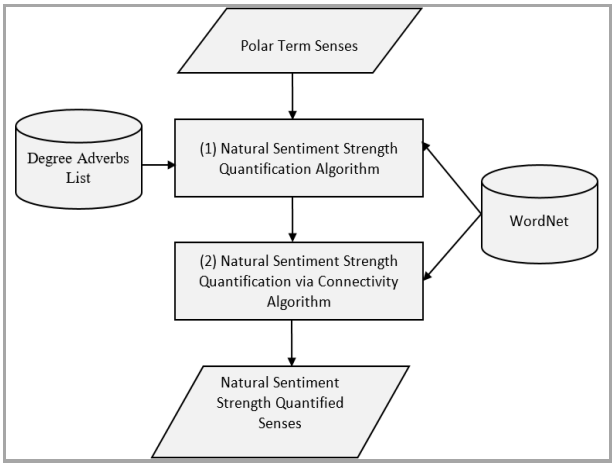


Figure 1. High level model architecture.

### B. Manual Compilation of Degree Adverbs List

An intensifier/diminisher is a lexical term that does not alter the original meaning of the term it modifies, but rather serves to intensify/diminish the strength (intensity) of the sentiment the term expresses [8], [27]-[30]. Intensifiers/diminishers are generally adverbs, often referred to as degree modifier words or degree adverbs in the literature. For example, the intensifier *extremely* in *extremely good movie* intensifies the positive sentiment of *good movie*, while the diminisher *barely* in *barely good movie* diminishes the positive sentiment of *good movie*.

Several research works [8], [27], [28] request for a group of annotators to manually assign coefficient values to degree adverbs, based on their degree to intensify or diminish the terms they modify. Following their procedure, in this work, a degree adverbs list was manually compiled for use by the model. This list was extracted based on the most-used adverbs in prior work, and comprises 19 intensifiers (very, really, extremely, etc.) and nine diminishers (slightly, barely, somewhat, etc.).

Three human annotators assigned a coefficient value in the range of 0.0 to 2.0 to each degree adverb in the list, based on the degree to which it intensifies or diminishes the term it modifies. The annotators are researchers in the field of Sentiment Analysis and Natural Language Processing, and were provided with a detailed explanation of the annotation requirements. The reason a range of 0.0 to 2.0 is chosen is because, according to [31], it is easier for annotators to assign a coefficient value in this range, since 0 can act as the weakest value, 1 as the mid-point, and 2 as the strongest value. For each degree adverb, several usage examples of the adverb in a context-of-use were presented to the annotator, so the annotator is able to have sufficient information about the degree adverb before estimating its correlation coefficient.

The final coefficient value for each degree adverb is the average value assigned by all annotators. For example, *somewhat* may be assigned a coefficient of 0.4, since it tends to greatly diminish polarity (*somewhat enjoyable*), while *very* may be assigned a coefficient of 1.6, since it tends to greatly intensify it (*very happy*). The full degree adverbs list utilized by the model is presented in Table I. The average coefficient value assigned by the three annotators is presented beside each degree adverb.

TABLE I. MANUALLY-COMPILED DEGREE ADVERBS LIST

| degree adverb | coefficient | degree adverb | coefficient |
|---|---|---|---|
| highly | 1.5 | pretty | 0.8 |
| very | 1.6 | apparently | 0.8 |
| extremely | 1.9 | seemingly | 0.8 |
| really | 1.6 | moderately | 0.5 |
| surpassingly | 1.8 | somewhat | 0.4 |
| particularly | 1.5 | questionably | 0.4 |
| excessively | 1.7 | slightly | 0.4 |
| strikingly | 1.7 | barely | 0.2 |
| extraordinarily | 1.9 | hardly | 0.2 |
| unusually | 1.6 | | |
| exceptionally | 1.6 | | |
| intensely | 1.8 | | |
| too | 1.5 | | |
| totally | 1.7 | | |
| utterly | 1.6 | | |
| perfectly | 1.6 | | |
| entirely | 1.6 | | |
| immensely | 1.8 | | |
| more | 1.5 | | |

### C. Natural Sentiment Strength Quantification Algorithm

Algorithm 1 presents the pseudocode for the Natural Sentiment Strength Quantification Algorithm (NSSQA). The NSSQA involves a fully semantic approach that utilizes only the gloss information and the manually-compiled list of degree adverbs to quantify the Natural Sentiment Strength (NSS) of polar term senses. In other words, the NSS of a term sense is quantified based on its context to appear with a degree adverb within the glosses of other senses.

**Natural Sentiment Strength Quantification Algorithm**

**Input:** lexicon, degree_adv_list
**Output:** NSS-quantified syns in pos_lex and neg_lex
**Begin**
**For** input_syn in lexicon:
 **For** gloss_term in gloss[input_syn]:
  **IF** match(gloss_term & degree_adv)
   **IF** NSS(modified_term) = null
    NSS(modified_term) = 1
   **End If**
   NSS(input_syn) = deg_adv_coefficient × NSS(modified_term)
   NSS(input_syn) propagation to syns with same term
   NSS(modified_term) propagation to syns with same term
  **End If**
 **End For**
**End For**
**End**

Algorithm 1. Natural sentiment strength quantification algorithm.

For each input synset input_syn that contains a degree adverb degree_adv in its gloss, the gloss term it modifies modified_term is automatically assigned an initial NSS of 1, under the condition it does not already have an NSS value assigned by a previous round of execution. If it does, then it retains its old NSS value. The coefficient value of the matched degree adverb deg_adv_coefficient is multiplied with the NSS value of the gloss term it modifies NSS(modified_term), resulting in a new NSS value NSS(input_syn) that is assigned to input_syn. NSS(input_syn) is then propagated to all synsets that share the same lemma term under the same POS category, while NSS(modified_term) is propagated to all synsets that share the same lemma term under the same POS category. This is repeated iteratively until there are no more possible synsets to assign with an NSS. The NSSQA is individually run for the list of positive synsets and the list of negative synsets in the lexicon.

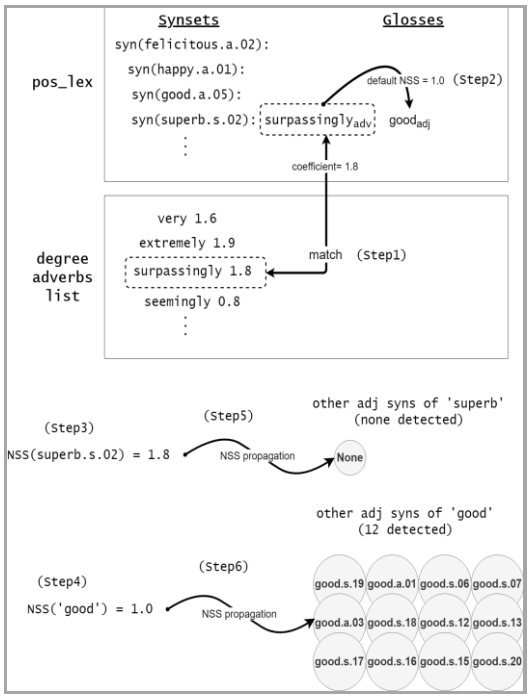


Figure 2. Example of the functionality of the NSSQA.

Fig. 2 depicts a detailed example of the functionality of the NSSQA during a match with the positive input synset superb.s.02. Since it has been naturally defined by humans as *surpassingly good*, most would agree that *superb* must have a higher strength compared to *good*, since it is *'surpassingly good'*, where *surpassingly* here plays the role of an intensifier.

In step one, there is a match between the degree adverb *surpassingly* and a term in the gloss of superb.s.02, and the coefficient value of *surpassingly*, 1.8, is retrieved. In step 2, the modified term *good* is assigned an initial default NSS of 1. In step 3, NSS(superb.s.02) is computed by degree_adv_coefficient × NSS(modified_term), which is 1.8 ×1 = 1.8. In step 4, NSS(good) is set as the default NSS of 1. In step 5, the NSS of superb.s.02 is propagated to all synsets that share the same lemma term, under the same POS; and in step 6, the NSS of the modified term *good* is propagated to all synsets that share the same lemma term, under the same POS. In this case, no propagation occurs for NSS(superb.s.02) = 1.8, since there are no synsets with a similar lemma term in the lexicon, but NSS(good) = 1 is propagated to a total of 12 synsets, including good.s.13, good.s.20, and good.s.19, as shown in the figure.

Finally, this generated a list of NSS-quantified synsets, by using gloss information and degree adverbs, as well as via NSS propagation. These NSS-quantified synsets are used as seed synsets by the next algorithm, in order to quantify the remaining synsets in the lexicon that have not yet been labeled with an NSS.

### *D. Natural Sentiment Strength Quantification via Connectivity Algorithm*

The Natural Sentiment Strength Quantification via Connectivity Algorithm (NSSQCA) involves using all the NSS-quantified synsets generated by the NSSQA as seed synsets, to label the remaining unlabelled synsets. Some synsets in the lexicon remain to be unlabelled with an NSS, since they do not contain degree adverbs within their glosses.

According to the Social Network Theory [32], a node that has many semantic relations with other nodes that are members of a particular category, also tends to be more 'central' to that category. Moreover, in the related literature, semantic connectivity between terms in a semantic network has been proven to be a reliable indicator of polarity [15]. Based on the Social Network Theory and on the concept of semantic connectivity, the NSS values of the remaining unlabelled synsets are quantified based on their connections with seed synsets in a semantic network. An input synset is assigned the average of all the NSS values of the seed synsets it matches with.

Algorithm 2 presents the pseudocode for the NSSQCA algorithm. For each input synset input_syn in the lexicon that is not yet labeled with an NSS, the neighbor synsets are extracted from the WordNet semantic network. The 'neighbor' of an input synset is defined as any synset connected to the input synset via the semantic relations of *similar-to*, *also-see*, *derivationally-related-form*, and *pertainym*, since these relations have been empirically

observed to optimally preserve sentiment properties during propagation in the semantic network.

**Natural Sentiment Strength Quantification via Connectivity Algorithm**

```
Input: -> seed_syns (NSS-quantified syns)
       -> remaining unlabeled syns in lexicon
Output: all syns in lexicon with final NSS
Begin
Function getNeighbors(syn_and_neighbors)
 For syn in syn_and_neighbors:
  syn_and_neighbors += syn.also_sees +
syn.similar_tos +
syn.derivationally_related_forms +
 syn.pertainyms
 End For
 return(syn_and_neighbors)
End getNeighbors
Function checkMatches(syn_and_neighbors)
 For syn in syn_and_neighbors:
  For seed in seed_syns:
   IF syn == seed:
    num_of_matches += 1
  End For
 End For
 return(num_of_matches)
End checkMatches
For input_syn in lexicon and not in seed_syns:
 syn_and_neighbors += input_syn
 syn_and_neighbors +=
getNeighbors(syn_and_neighbors)
 semantic_connections_iters = 0
 For x in range(6):
  num_of_matches =
checkMatches(syn_and_neighbors)
  If num_of_matches == 0:
   syn_and_neighbors +=
getNeighbors(syn_and_neighbors)
   semantic_connections_iters += 1
  End If
 End For
 penalty_score = 0.1 × semantic_connections_iters
 NSS(input_syn) = NSS_values_of_matched_seeds /
num_of_matches
 NSS(input_syn) = NSS(input_syn) – penalty_score
End For
End
```

Algorithm 2. Example of the functionality of the NSSQA.

The motivation underlying the utilization of neighbors is that their semantic properties are preserved, and all neighbor synsets connected to the input synset via these relations are assumed to carry the same NSS value as the input synset itself. In this manner, any synset connected to the input synset via the mentioned semantic relations is considered to be a neighbor of the input synset, and is employed to aid in the matching process.

Next, each syn in syn_and_neighbors (the set comprising the input synset as well as its neighbors) is compared to all the seed synsets seed_syns to check for any possible matches. If there is no match detected after the default initial matching step, the syn_and_neighbors is iteratively expanded by adding new neighbors for each syn in syn_and_neighbors, with each additional iteration of semantic connections.

It has been well-established in the literature [33], [34] that the semantic significance of a path decreases as a function of its length from the source synset to the destination synset in the WordNet semantic network. In other words, the semantic relationship between the two synsets becomes weaker as the path between them increases. Reference [21] demonstrates that semantic connections over six iterations become 'noisy', i.e., the semantic relations become very weak, and the sentiment properties are no longer effectively preserved.

Therefore, a maximum of six iterations of semantic connections is performed by this algorithm. On every iteration, the algorithm checks for possible matches between the expanded syn_and_neighbors and the seed_syns. If it does not detect any matches with seeds, it performs an additional iteration of semantic connections, and the possibility of a match is increased, with the availability of a now larger set of semantic connections to aid in the matching process. It terminates either when at least one match is found, or until it has reached a maximum of six iterations. If there are no matches detected between syn_and_neighbors and seed_syns after six iterations, the input_syn is assigned an NSS of 0, and is assumed to carry no polarity.

Since the semantic relationship becomes weaker with each additional semantic connections iteration, the relationship between the sentiment properties of the two synsets also becomes weaker. Therefore, a penalty score ($P$) is employed to effectively dampen the final NSS of input synsets that have matches with seeds via semantic connections, over the initial default iteration. The value of $P$ is increased as the number of semantic connections iterations increases, and can be defined as follows:

$$P = C \times iters \tag{1}$$

where $C$ is some constant (heuristically set as 0.1), and *iters* represents the number of iterations of semantic connections (in the range of 1-6).

The final NSS value of input_syn is computed as the average of the NSS values of all the seeds it has matched with, as follows:

$$NSS(input\_syn) = \frac{NSS\_values\_of\_matched\_seeds}{num\_of\_matches} \tag{2}$$

Finally, the final NSS value is updated after applying $P$, as follows:

$$NSS(input\_syn) = NSS(input\_syn) - P \tag{3}$$

The NSSQCA is individually run for the list of positive synsets and the list of negative synsets in the lexicon.

Fig. 3 depicts a detailed example of the functionality of the NSSQCA for the input synset record-breaking.s.01. The initial default 'semantic connections' are retrieved via the mentioned semantic relations, so syn_and_neighbors = {record-breaking.s.01, best.s.01}. Since there are no matches detected between syn_and_neighbors and seed_syns, the algorithm expands syn_and_neighbors by an additional semantic connections iteration (semantic_connections_iters = 1). syn_and_neighbors now

contains 17 synsets. The algorithm terminates after the first iteration, since two matches are now detected between syn_and_neighbors and seed_syns. With this, NSS(record-breaking.s.01) = 1.3, and after applying the penalty score for one semantic connections iteration, the NSS is re-updated to 1.2, as follows:

$$NSS(record\text{-}breaking.s.01) = \frac{NSS(good.a.01) + NSS(superfine.s.03)}{num\_of\_matches} \quad (4)$$

$$= \frac{1+1.6}{2} = 1.3$$

$$P = C \times iters = 0.1 \times 1 = 0.1$$

$$NSS(record-breaking.s.01) = NSS(record-breaking.s.01) - P$$
$$= 1.3 - 0.1$$
$$= 1.2$$

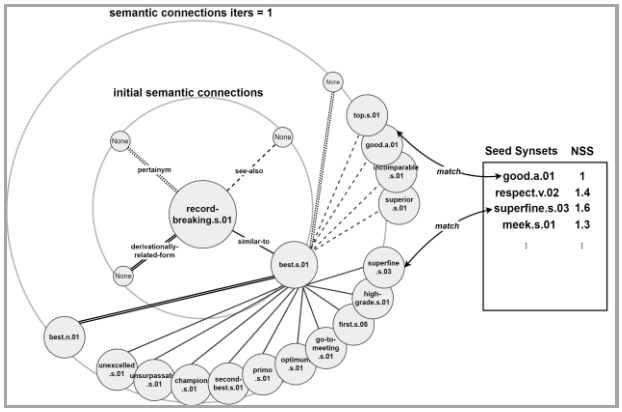


Figure 3. Example of the functionality of the NSSQCA.

The NSSQCA assigns all the remaining unlabeled synsets in the lexicon in this manner. The NSS values generated are in the range of 0 to 2. Therefore, in the final generated sentiment lexicon, each NSS value is divided by 2 to normalize all values to within the range of 0 to 1, as a measure of the final NSS for each of the synsets in the lexicon. For example, NSS(record-breaking.s.01) = 1.2/2 = 0.6, which represents its final NSS value within the generated lexicon. The intuition underlying this normalization is that related lexicons that contain strength values also use a range of 0-1 [21].

## IV. Experimental Setup and Evaluation

The experimental setup and evaluation procedure involves computing the model's overall accuracy in quantifying the NSS of subjective senses, and then comparing it to that of related models on the same gold standard benchmarks. The only available sense-level lexicon generation model that considers strength values is the SentiWordNet 3.0 lexicon generation model [21]. Other sense-level lexicon generation models by [19] and [20] do not consider sentiment strength.

Note that, although the work of [21] to generate SentiWordNet 3.0 is a relatively early work, it is to date a widely applied and popular sense-level lexicon in this area, and has been accepted as reliable in practical application. Therefore, recent work by [19] considers it 'state-of-the-art' and provides a comparison to this important work. Following this, the proposed model is also compared to the SentiWordNet 3.0 lexicon generation model.

Reference [16] requests for a group of human annotators to rate a set of terms extracted from their generated sentiment lexicon in the range of [0, 1]. The average value from the values given by all annotators represented the final sentiment strength for each term. This manually-annotated benchmark was then used to measure their semantic distance-based sentiment lexicon generation model. They only use a portion of the terms from the generated lexicon, and have human annotators manually-annotate these terms with sentiment strength values. They then compute the accuracy of the model to label the terms against this manually- annotated benchmark, and generalize this as the overall accuracy of the model to assign terms with sentiment strength values, since it is extremely time-consuming and tedious for annotators to manually annotate the entire generated sentiment lexicon.

Following this procedure, 300 positive synsets and 300 negative synsets were randomly extracted from the generated sentiment lexicon. Three human annotators skilled in linguistics and natural language processing were requested to rate each of the synsets in the range of [0, 1]. The average value from the values given by all annotators represents the sentiment strength for each synset. For each synset presented to the annotators, its formal definition and multiple examples of the synset in a context of use were also displayed. This is so the annotator can have sufficient information about the synset before estimating its sentiment strength.

The correlation coefficient (correl) between the NSS values of the synsets in the generated lexicon, and the strength values of the synsets in the manually-annotated benchmark, are taken as representative of the overall accuracy of the model to label subjective term senses with a sentiment strength. Correl is a statistical measure that measures the strength of the relationship between the NSS values generated by the model and the manually-annotated NSS values, and can be computed as follows:

$$correl = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} - \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (5)$$

where $x$ represents the values generated by the proposed model, while $y$ represents the values manually defined by human annotation. In order to allow for a feasible comparison to the SentiWordNet 3.0 lexicon generation model, the correlation coefficient between the strength values of the synsets in the SentiWordNet lexicon, and the strength values of the synsets in the manually-annotated benchmark, are taken as representative of the overall accuracy of the SentiWordNet model to label subjective term senses with a sentiment strength. Each synset in the SentiWordNet lexicon has both a positive and a negative

score, simultaneously. For a match between a synset in the SentiWordNet lexicon and the positive list in the manually-annotated benchmark, the positive SentiWordNet score for that synset is used as its sentiment strength, while for a match between a synset in the SentiWordNet lexicon and the negative list in the manually-annotated benchmark, the negative SentiWordNet score for that synset is used as its sentiment strength.

## V. Results and Discussion

### A. Output Results by the NSSQA

The Natural Sentiment Strength Quantification Algorithm (NSSQA) involves a fully semantic approach that utilizes only the gloss information and a list of predefined degree adverbs to quantify the Natural Sentiment Strength (NSS) of polar term senses. The sense-level sentiment lexicon lexicon used as an input into this model was manually-compiled by three human annotators. From a set of random WordNet senses, the annotators labeled approximately 8,101 synsets as positive and 3,186 synsets as negative. The reason there is a higher number of positive synsets is because the set of WordNet synsets used for annotation contained more positive synsets to choose from, as compared to negative.

After running the NSSQA, Table II presents the coverage results. The total number of NSS-quantified synsets is 289. Note that the total number of NSS-quantified modified terms is also 289, since on each step of the algorithm, the modified term is assigned a default NSS of 1, and the input synset is assigned an NSS based on the coefficient value of the degree adverb multiplied by the NSS of the modified term. Therefore, in the same step, both are simultaneously quantified with an NSS. However, NSS-quantified modified terms are not considered in the coverage results, since they are not synsets, and their only function is to aid in the propagation of their NSS-values to other synsets available in the lexicon.

The total synsets quantified via the NSS propagation step is 314. This involves the NSS values from the NSS-quantified synsets and the NSS-quantified modified terms being propagated to other synsets in the lexicon that have the same lemma term, provided they are of the same POS. For example, fabulous.s.01 is quantified with an NSS of 1.9, and this is propagated to other adjective synsets in the lexicon that share the same lemma term. In this case, fabulous.s.02 is detected in the lexicon, but does not yet have an NSS, so the NSS of 1.9 is propagated from fabulous.s.01 to fabulous.s.02.

TABLE II. Coverage Results by NSSQA

| NSS-quantified synsets | pos | neg | all |
|---|---|---|---|
| by gloss degree adverbs matching | 206 | 83 | 289 |
| via NSS propagation | 216 | 98 | 314 |
| **Total** | 422 | 181 | 603 |

### B. Output Results by the NSSQCA

Based on the NSS-quantified synsets (289) and the NSS-quantified synsets via NSS propagation (314) as 'seed synsets', the remaining unlabeled synsets in the lexicon are assigned NSS values. After running the NSSQCA, Table III shows some senses (adjectives, adverbs, nouns and verbs) from the generated lexicon, distributed according to the NSS range they fall under.

As shown by the samples, the NSSQCA is able to successfully quantify the NSS values of polar senses on a fine-grained scale in the range of [0, 1]. For example, the algorithm successfully assigns a low NSS to senses with a weak sentiment strength (NSS(acceptable.a.01) = 0.4), and a high NSS to senses with a strong sentiment strength (NSS(fantastic) = 0.95). This lexicon is made publically available for research purposes[1].

TABLE III. Sample Senses from Generated Lexicon Distributed by NSS Range

| NSS | Positive Senses | Negative Senses |
|---|---|---|
| **0.9-1.0** | fantastic.s.02<br>fabulous.s.01<br>magnificence.n.02<br>beautify.v.01 | afraid.a.01<br>nerve-racking.s.01<br>disgustingness.n.02<br>fear.v.02 |
| **0.7-0.8** | superfine.s.03<br>respectably.r.02<br>excellence.n.01<br>idolize.v.01 | stupid.a.01<br>dispassionately.r.01<br>selfishness.n.01<br>grouch.v.01 |
| **0.5-0.6** | record-breaking.s.01<br>honestly.r.02<br>pleasure.n.01<br>adore.v.01 | ill-natured.s.01<br>displeasingly.r.01<br>recklessness.n.01<br>abandon.v.05 |
| **0.3-0.4** | acceptable.a.01<br>graciously.r.01<br>highness.n.02<br>pride.v.01 | ashamed.a.01<br>disloyally.r.01<br>falsification.n.04<br>pollute.v.01 |
| **0.1-0.2** | straight.a.02<br>justifiably.r.01<br>guidance.n.02<br>align.v.04 | disrupted.s.01<br>sporadically.r.01<br>hardness.n.01<br>scarify.v.01 |

### C. Evaluation Results against Manually-Annotated Benchmark

Table IV presents the evaluation results of the model's overall accuracy in quantifying the NSS of polar term senses. The correlation coefficient between the NSS values of the synsets in the generated lexicon, and the NSS values of the synsets in the manually-annotated benchmark, are taken as representative of the overall accuracy of the model to label subjective term senses with a sentiment strength.

The final correlation coefficient achieved by the model is 0.694 and 0.676 for the positive and negative categories, respectively, yielding an overall accuracy of 0.685. Based on these results, the model performs with similar accuracy in the NSS quantification of both positive and negative term senses.

The only sense-level lexicon generation model available for comparison is the SentiWordNet 3.0 generation model [21]. As shown in Table IV, the final correlation coefficient achieved by the SentiWordNet lexicon generation model is 0.435 and 0.303 for the positive and

[1] http://www.ftsm.ukm.my/NSSsentimentlexicon.zip

negative categories, respectively, yielding an overall accuracy of 0.369.

TABLE IV. EVALUATION RESULTS OF MODEL'S OVERALL ACCURACY

| Model | pos | neg | overall |
|---|---|---|---|
| proposed model | 0.694 | 0.676 | 0.685 |
| SentiWordNet 3.0 model | 0.435 | 0.303 | 0.369 |

To allow for a side-by-side comparison, Fig. 4 depicts the accuracy plots on a graph for the proposed model vs the SentiWordNet 3.0 model. As evident in the figure, the proposed model outperforms the SentiWordNet 3.0 model by a large margin (0.316). This demonstrates that the SentiWordNet model fails to quantify the sentiment strength that subjective senses carry, most likely due to the 'artificially statistical' nature inherent in its approach.

The strength values generated by their model tend to reflect the 'likeliness' of a word being in a particular class, or the confidence in the accuracy of labeling, rather than the actual sentiment strength of the sense. Recalling that their model employs a committee of eight supervised classifiers, the assignment of senses with strength values purely relies on *statistical means*, and there is no *semantic mechanism* that explicitly measures the strength of senses in their work.

As a solution, the proposed model involves a fully semantic approach that utilizes only the human-defined gloss information, a semantic network, and degree adverbs, to quantify the truly 'natural' sentiment strength of term senses. The model has successfully addressed the issues inherent in the popular SentiWordNet 3.0 generation model in terms of its ability to generate strength values to term senses, as demonstrated by the superior accuracy achieved by the model in comparison to that achieved by the SentiWordNet 3.0 model. This validates its practical application in reliably quantifying the NSS values of any readily-available term- or sense-level polarity lexicons.

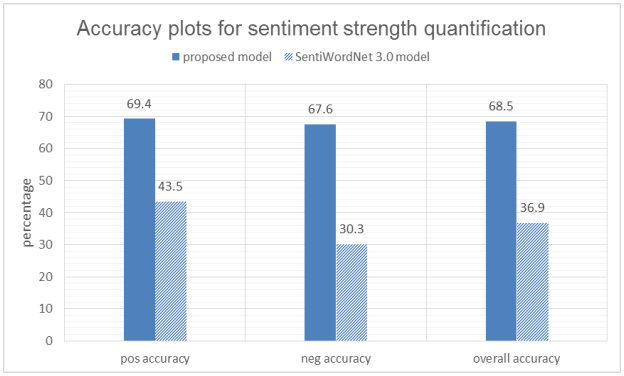


Figure 4. Accuracy graph for natural sentiment strength quantification.

## VI. CONCLUSION AND FUTURE WORK

In contrast to prior models that rely on statistical means to measure strength and in turn generate biased results, this model utilizes only the semantic information manually encoded within the human-defined glosses of term senses, a semantic network, and a small set of manually-compiled degree adverbs, in order to quantify the 'Natural' Sentiment Strength (NSS) of senses.

Evaluation of this model to accurately assign senses with NSS value demonstrates its superior performance when compared to state-of-the-art models, against the same gold standard benchmarks. This work has practical implications in automatically labelling readily-available term- and sense-level polarity lexicons with NSS information.

In future work, we plan to apply the proposed algorithms on other languages [35], [36]. The generated lexicon can also be assessed in practical sentiment analysis applications in various domains, e.g., recommender systems [37], [38], movie reviews [39], social media content, etc. We also plan to investigate the contribution to accuracy of the integration of a corpus [40], in the process of assigning a natural sentiment strength to term senses. The context surrounding the term can be utilized to aid in the assignment process.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

Mohammad Darwich has designed and developed the algorithms. All of the researches have worked together to validate the results obtained, and approve the final version of this paper.

## ACKNOWLEDGMENT

This research was partially supported by the Malaysia Ministry of Education Grant FRGS/2/2013/ICT02/UKM/02/1 awarded to the Center for Artificial Intelligence Technology at Universiti Kebangsaan Malaysia.

## REFERENCES

[1] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. Eighth International AAAI Conference on Weblogs and Social Media*, 2014, pp. 216-225.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proc. the Workshop on Language in Social Media*, 2011, pp. 30-38.

[3] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proc. the 23rd International Conference on Computational Linguistics: Posters*, 2010, pp. 241-249.

[4] M. Thelwall, "The heart and soul of the web? Sentiment strength detection in the social web with SentiStrength," in *Cyberemotions*, Springer, 2017, pp. 119-134.

[5] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544-2558, December 2010.

[6] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 115-124.

[7] T. Wilson, *et al.*, "OpinionFinder: A system for subjectivity analysis," in *Proc. HLT/EMNLP Interactive Demonstrations*, 2005, pp. 34-35.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267-307, 2011.
[9] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," arXiv Preprint arXiv:1103.2903, 2011.
[10] A. Schneider, J. Male, S. Bhogadhi, and E. Dragut, "DebugSL: An interactive tool for debugging sentiment lexicons," in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2018, pp. 36-40.
[11] E. Dragut, H. Wang, C. Yu, P. Sistla, and W. Meng, "Polarity consistency checking for sentiment dictionaries," in *Proc. the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, 2012, vol. 1, pp. 997-1005.
[12] S. M. Mohammad and F. Bravo-Marquez, "Emotion intensities in tweets," arXiv Preprint arXiv:1708.03696, 2017.
[13] R. Sharma, A. Somani, L. Kumar, and P. Bhattacharyya, "Sentiment intensity ranking among adjectives using sentiment bearing word embeddings," in *Proc. Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 547-552.
[14] W. Peng and D. H. Park, "Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization," in *Proc. Fifth International AAAI Conference on Weblogs and Social Media*, 2011, pp. 273-280.
[15] A. Hassan and D. Radev, "Identifying text polarity using random walks," in *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 395-403.
[16] G. K. Williams and S. S. Anand, "Predicting the polarity strength of adjectives using wordnet," in *Proc. Third International AAAI Conference on Weblogs and Social Media*, 2009, pp. 346-349.
[17] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. Reis, and J. Reynar, "Building a sentiment summarizer for local service reviews," in *Proc. Www Workshop: Nlp in the Information Explosion Era*, 2008.
[18] A. Adreevskaia and S. Bergler, "Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses," in *Proc. 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006, pp. 209-216.
[19] I. S. Vicente, R. Agerri, and G. Rigau, "Q-wordnet ppv: Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages," arXiv Preprint arXiv:1702.01711, 2017.
[20] R. Agerri and A. Garc á-Serrano, "Q-WordNet: Extracting polarity from WordNet senses," in *Proc. LREC*, 2010, pp. 2300-2305.
[21] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. LREC*, 2010, vol. 10, pp. 2200-2204.
[22] D. M. E. D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University-Engineering Sciences*, vol. 30, pp. 330-338, October 2018.
[23] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102-107, March 2016.
[24] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42-49, 2016.
[25] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 5876-5883.
[26] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proc. World Wide Web Conference*, 2018, pp. 1165-1174.
[27] M. Schulder, M. Wiegand, J. Ruppenhofer, and S. Köser, "Introducing a lexicon of verbal polarity shifters for English," in *Proc. LREC*, 2018, pp. 1393-1397.
[28] S. Kiritchenko, S. Mohammad, and M. Salameh, "Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases," in *Proc. the 10th International Workshop on Semantic Evaluation*, 2016, pp. 42-51.
[29] E. Dragut and C. Fellbaum, "The role of adverbs in sentiment analysis," in *Proc. of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, 2014, pp. 38-41.
[30] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110-125, May 2006.
[31] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "Textual affect sensing for sociable and expressive online communication," in *Proc. International Conference on Affective Computing and Intelligent Interaction*, 2007, pp. 218-229.
[32] R. S. Burt, "Models of network structure," *Annual Review of Sociology*, vol. 6, no. 1, pp. 79-141, 1980.
[33] N. Ide, "Making senses: Bootstrapping sense-tagged lists of semantically-related words," in *Proc. International Conference on Intelligent Text Processing and Computational Linguistics*, 2006, pp. 13-27.
[34] J. Wiebe and R. Mihalcea, "Word sense and subjectivity," in *Proc. the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006, pp. 1065-1072.
[35] M. Darwich, S. A. M. Noah, and N. Omar, "Minimally-supervised sentiment lexicon induction model: A case study of malay sentiment analysis," in *Proc. International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, 2017, pp. 225-237.
[36] M. Darwich, S. A. M. Noah, and N. Omar, "Automatically generating a sentiment lexicon for the Malay language," *Asia-Pacific Journal of Information Technology and Multimedia*, vol. 5, no. 1, pp. 49-59, 2016.
[37] N. A. Osman, S. A. M. Noah, and M. Darwich, "Contextual sentiment based recommender system to provide recommendation in the electronic products domain," *International Journal of Machine Learning and Computing*, vol. 9, no. 4, pp. 425-431, 2019.
[38] N. A. Osman and S. A. M. Noah, "Sentiment-Based model for recommender systems," in *Proc. Fourth International Conference on Information Retrieval and Knowledge Management*, 2018, pp. 1-6.
[39] I. S. Ahmad, A. A. Bakar, and M. R. Yaakub, "A review of feature selection in sentiment analysis using information gain and domain specific ontology," *International Journal of Advanced Computer Research*, vol. 9, no. 44, pp. 283-292, 2019.
[40] M. Darwich, S. A. M. Noah, N. Omar, and N. A. Osman, "Corpus-Based techniques for sentiment lexicon generation: A review," *Journal of Digital Information Management*, vol. 17, no. 5, pp. 296-305, 2019.



**Mohammad Darwich** has received his master's degree in information technology from the National University of Malaysia in 2014. His research interests comprise textual sentiment analysis and emotion analysis in particular, and natural language processing and text mining in general. He is currently completing his PhD in computer science at the National University of Malaysia.

**Shahrul Azman Mohd Noah** received his MSc and PhD degrees in information studies from Sheffield University. He is a professor in the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia and currently heads the Knowledge Technology research group. His current research work is focused on semantic computing with special emphasis on information retrieval, ontology and recommender systems.

**Nazlia Omar** received the B.Sc. degree (Hons.) from UMIST, U.K, the M.Sc. degree from the University of Liverpool, U.K., and the Ph.D. degree from the University of Ulster, U.K. She is currently an Associate Professor with the Center for AI Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. Her main research interest is in the area of natural language processing and computational linguistics.

**Nurul Aida Osman** has received her master's degree in information technology from Universiti Kebangsaan Malaysia in 2012. Her current research work is focused on sentiment analysis and recommender systems. Her other research interest is also in the area of knowledge engineering and ontology. She is currently completing her PhD in computer science from Universiti Kebangsaan Malaysia.

**Ibrahim Said Ahmad** is a lecturer in Department of Information Technology, Bayero University Kano. He received his BSc degree in Computer Science from Bayero University Kano, Nigeria, in 2011 and his MSc degree in Information Technology from The University of Nottingham, UK, in 2014. He is currently a PhD candidate in Universiti Kebangsaan Malaysia. He has published a number of journal articles and attended many conferences.