A Meta-Framework for Modeling the Human Reading Process in Sentiment Analysis

RAMY BALY, ROULA HOBEICA, HAZEM HAJJ, and WASSIM EL-HAJJ,
American University of Beirut
KHALED BASHIR SHABAN, Qatar University
AHMAD AL-SALLAB, Cairo University

This article introduces a sentiment analysis approach that adopts the way humans read, interpret, and extract sentiment from text. Our motivation builds on the assumption that human interpretation should lead to the most accurate assessment of sentiment in text. We call this automated process Human Reading for Sentiment (HRS). Previous research in sentiment analysis has produced many frameworks that can fit one or more of the HRS aspects; however, none of these methods has addressed them all in one approach. HRS provides a meta-framework for developing new sentiment analysis methods or improving existing ones. The proposed framework provides a theoretical lens for zooming in and evaluating aspects of any sentiment analysis method to identify gaps for improvements towards matching the human reading process. Key steps in HRS include the automation of humans low-level and high-level cognitive text processing. This methodology paves the way towards the integration of psychology with computational linguistics and machine learning to employ models of pragmatics and discourse analysis for sentiment analysis. HRS is tested with two state-of-the-art methods; one is based on feature engineering, and the other is based on deep learning. HRS highlighted the gaps in both methods and showed improvements for both.

CCS Concepts: • **Information systems → Retrieval tasks and goals;**

Additional Key Words and Phrases: Sentiment analysis, human reading, psychology, supervised learning and notions

ACM Reference Format:
Ramy Baly, Roula Hobeica, Hazem Hajj, Wassim El-Hajj, Khaled Bashir Shaban, and Ahmad Al-Sallab. 2016. A meta-framework for modeling the human reading process in sentiment analysis. ACM Trans. Inf. Syst. 35, 1, Article 7 (August 2016), 21 pages.
DOI: http://dx.doi.org/10.1145/2950050

1. INTRODUCTION

Sentiment analysis is a process by which useful sentiment-related knowledge can be deduced from data. According to Liu and Zhang [2012], research in sentiment analysis spans many sub-tasks, including sentiment classification in documents, sentences and microblogs, aspect-based sentiment analysis, identifying opinion holders and targets,

Authors' addresses: R. Baly, R. Hobeica, and H. Hajj, Electrical and Computer Engineering Department, American University of Beirut, Beirut, Lebanon; emails: rgb15@aub.edu.lb, roula.hobeica@gmail.com, hh63@aub.edu.lb; W. El-Hajj, Computer Science Department, American University of Beirut, Beirut, Lebanon; email: we07@aub.edu.lb; K. B. Shaban, Computer Science and Engineering Department, Qatar University, Doha, Qatar; email: khaled.shaban@qu.edu.qa; A. Al-Sallab, Cairo University, Cairo, Egypt; email: ahmad.elsallab@gmail.com.

DOI: http://dx.doi.org/10.1145/2950050

and opinion spam detection. The amount of online subjective data has been increasing exponentially in forums, social networks, and personal blogs [Chen and Zimbra 2010]. As an example, Facebook gets 500 Terabytes of new data per day [Hendler 2013].

The web presents a rich source of sentiment-related data, including personal statements and comments on life matters, products, and so on. Sentiment analysis applications span across multiple domains, such as politics and business, and provide insight into public opinion regarding policies, products, or popular items. In politics, sentiment analysis was employed at the onset of the 2012 United States presidential elections to show who was leading in public popularity [Hoffman 2013]. Sentiment analysis was also used to model people's behavior at the onset of major events, such as the death of Steve Jobs, which caused a "sad" spike of sentiment on Twitter [Rawkes 2011]. On the financial side, sentiment analysis systems can provide businesses with key insights for efficient market strategies. These examples of getting access into people's opinions have fueled research in sentiment analysis since the mid-2000s. In this article, the focus of the sentiment analysis task is on the derivation of sentiment orientation at the document level. This focus is motivated by the availability of large amounts of opinionated documents on the web.

In the context of document sentiment classification (DSC), research has covered several problems including feature extraction [Farra et al. 2010], feature selection [Abbasi et al. 2011], and supervised machine-learning algorithms [Dang et al. 2010]. Other research that has reported high levels of accuracy in DSC include Yessenalina et al. [2010], Glorot et al. [2011], Tu et al. [2012], Le and Mikolov [2014], Tang et al. [2015b], and Tang et al. [2015a]. These methods will be further discussed and analyzed in the article.

Predicting sentiment with high accuracy has been the main and most common motivation to all researchers, and the proposed models achieved accuracies ranging between 74.39% [Turney 2002] and 90.7% [Abbasi et al. 2011]. We conjecture that the highest accuracies can be achieved by methods that come closest to the human reading process, which this article is proposing to automate. While previous work has looked at matching human aspects of text interpretation, these approaches were limited to direct employment of natural language processing (NLP) and machine-learning (ML) approaches. To the best of our knowledge, there have not been attempts to take a fundamental approach in matching the human process of extracting sentiment from text. We propose to mimic the human reading process for the purpose of DSC. The proposed model builds its foundations from the field of psychology by understanding the process by which humans read and interpret sentiment from text and developing the algorithms for automating this process. The aim is to develop a human-inspired system that takes a document as input and predicts its sentiment.

We propose a meta-framework that formulates sentiment analysis approaches in a human-like framework for automated DSC, and we name it "Human Reading for Sentiment" (HRS). Key steps in HRS include the automation of the humans' cognitive text processing. We describe the human reading process as depicted in psychology [Grabe and Stoller 2013] and propose equivalent automated steps in HRS towards achieving highest possible accuracies. We emphasize a central idea to HRS, which is the capture of notions [Hobeica et al. 2011] that include both textual representations and sentiment polarities. We provide a comprehensive coverage of the concept of notions by including an expanded definition and proposing additional human-related features such as working memory and notions synonyms.

The main contribution of this article is the development of a meta-framework, a generic description that enables the development of sentiment analysis frameworks that are complete in their representation of the human reading process. The proposed meta-framework can be used to extend existing sentiment analysis methods or as a

reference when developing new methods from scratch. In this article, we use HRS to extend two state-of-the-art methods originating from two schools of ML: feature engineering and deep learning. The resulting frameworks are described and evaluated showing the improvements achieved from the meta-framework.

The rest of the article is organized as follows. Section 2 reviews related work in the field of DSC. Section 3 presents the psychological foundation of the HRS. This section also motivates the need for the HRS framework by identifying gaps with previous methods. Section 4 describes the proposed steps to automate the HRS model. Section 5 describes the experiments with state of the art and reports the derived improvements. Section 6 concludes the article and presents suggestions for future work.

2. RELATED WORK

Since the mid-2000s, there has been an extensive focus on DSC. The main distinction between the different approaches was in the type and complexity of information that were used as features to infer sentiment. These features varied to cover surface, syntactic, and deeper semantics.

Surface features, referred to as bag-of-words (BoW) or word n-grams, were among the earliest features to be used for sentiment classification. Pang et al. [2002] proposed word n-grams to train different classifiers including Support Vector Machines (SVM), Naïve Bayes (NB), and Expectation Maximization (EM). Word n-grams features were also proposed to identify hidden sentiment factors in reviews and for product sales prediction [Yu et al. 2012]. In general, surface features provide a shallow and insufficient representation of the text semantics.

The establishment of NLP tools such as stemmers, part-of-speech taggers, and syntactic parsers has made it possible to use or incorporate syntactic information as additional features for sentiment classification to reflect the principles by which sentences are constructed. Predefined phrase patterns were proposed for sentiment analysis in consumer reviews, where each pattern was assigned sentiment scores using pointwise mutual information with "excellent" and "poor" keywords [Turney 2002]. Among different choices of linguistic preprocessing, stemming achieved best results when applied to bigrams and trigrams [Dave et al. 2003]. Stemming was also applied to create a new lexicon from the Multi-Perspective Question Answering (MPQA) lexicon [Wiebe et al. 2005] and the appraisal lexicon [Whitelaw et al. 2005]. The resulting lexicon was used by Lin et al. [2012] to perform unsupervised sentiment analysis in different reviews corpora. Abbasi et al. [2008] explored the impact of combining different types of syntactic and stylistic (surface) features. Syntactic features included: word and part-of-speech (POS) n-grams, whereas stylistic features included letter frequencies, character and digits n-grams, function words, and word length. Experiments using SVM showed that syntactic features achieve higher accuracy than stylistic features and that the union of both achieved the highest performance.

The development of semantic and sentiment lexica enabled the inclusion of deeper semantic information into sentiment analysis models. Among the most widely used English lexica are WordNet [Fellbaum 1999] and SentiWordNet [Esuli and Sebastiani 2006]. WordNet is a lexicon that provides semantic relations among the words by grouping them into synonym sets, each expressing a distinct concept [Fellbaum 1999]. SentiWordNet is a lexicon that assigns to each WordNet synonym set (synset) three scores representing its degree of positivity, negativity, and neutrality [Esuli and Sebastiani 2006].

Semantic features were introduced by associating n-grams with corresponding scores derived from SentiWordNet [Dang et al. 2010]. These features were used along with POS tags of adjectives, adverbs, and verbs to train SVM models for sentiment classification in consumer reviews. Abbasi et al. [2011] proposed a rich set of heterogeneous

n-gram features including surface features (character and word *n*-grams), syntactic features (POS, word-POS *n*-grams, and phrase patterns), and semantic features (word *n*-grams with words replaced by WordNet synset labels). Due to the huge size of the proposed features set, and in order to eliminate irrelevant and redundant ones, feature relation networks (FRN) was proposed to produce a smaller and more representative set [Abbasi et al. 2011]. The application of FRN achieved high accuracies when tested on product and movie reviews. Katakis et al. [2014] proposed a collection of semantic features that capture the number, type, and polarity score of terms and phrases that contribute to the sentence sentiment. Examples include the count of nouns with positive polarity and the count of verbal phrases with negative polarity. These features were used along with word *n*-grams to evaluate different ML classifiers on movie reviews. Tu et al. [2012] proposed to use only the subjective parts of a review to infer its sentiment. These parts were identified as syntactic sub-structures in parse trees that contain polarity terms extracted from sentiment lexicons.

Besides DSC, semantic features were also reported to achieve very high performances in the SemEval shared task on sentiment analysis in Twitter [Rosenthal et al. 2014]. The choice of features included characters, words and lemmas *n*-grams, tweet-specific features such as emoticons and abbreviations, and sentiment scores obtained form MPQA, SentiWordNet, and Bing Liu's sentiment lexica [Ding et al. 2008]. Mohammad et al. [2013b] proposed a set of hand-crafted features including the following: number of all-caps words, existence of emoticons with location, frequency of elongated words, number of negations and punctuation, word *n*-grams, and lexical features extracted from several sentiment lexicons including HL [Hu and Liu 2004], MPQA, Sentiment140 [Mohammad et al. 2013a] and the one developed by the National Research Council (NRC) [Mohammad and Turney 2010]. These hand-crafted features were augmented by sentiment-specific word embedding features learned by deep learning that include a language-model score and a sentiment score assigned to every word *n*-gram [Tang et al. 2014]. The framework proposed by Miura et al. [2014] included text normalizer, spelling corrector, POS tagger, word sense disambiguator, and rule-based negation detector. These components were used to extract character and word *n*-grams, lexical features from multiple lexica, and the different senses of every word, which are then used to train a logistic regression classifier.

Deep learning has emerged as the current state-of-the-art technique for sentiment analysis. Glorot et al. [2011] applied Stacked Denoising Autoencoders to word *n*-grams vectors in order to predict the reviews sentiment in a large corpus of Amazon reviews. Le and Mikolov [2014] proposed to derive document vector representations such that these vectors model the semantics distributed over all words in the document. This approach achieved high results when tested on movie reviews. Recently, hierarchical approaches were proposed for DSC by first combining word vectors to derive sentence-level representations and then using these representations to derive a document-level representation. Tang et al. [2015a] proposed the use of Convolutional Neural Networks (CNNs) for the first step and an average pooling layer for the second step. The resulting document vector was then added to other vectors that model author and product information available in the corpus. Tang et al. [2015b] also used CNN for the first step of the hierarchy, while proposing the use of Gated Recurrent Neural Networks (GRNNs) to model dependencies between sentences when deriving the document-level representation. In addition to DSC, a family of recursive deep learning approaches, namely Recursive Neural Networks (RNNs) [Socher et al. 2011] and Recursive Neural Tensor Networks (RNTNs) [Socher et al. 2013], has proven successful at modeling sentiment of different sentence constituents, ranging from words up to full sentences, in a bottom-up fashion following the structure of the syntactic parse trees. Overall, deep learning models proved to outperform SVM, NB, and bigram-based NB when applied to large sets of movie and restaurant reviews.

Table I. Reported Accuracies of the Proposed Approaches Mentioned in the Literature

Type	Features	Corpus	Accuracy
Surface	*n*-grams [Pang et al. 2002]	Movie reviews	82.9%
Syntactic	Predefined phrase patterns [Turney 2002]	Product reviews	74.4%
	Stemmed sentiment lexicons [Lin et al. 2012]	Product reviews	77.7%
	Stemmed *n*-grams [Dave et al. 2003]	Product reviews	83%
	Stylistic and syntactic features [Abbasi et al. 2008]	Movie reviews	88.04%
Semantic	Number, type and polarity of terms and phrases [Katakis et al. 2014]	Product reviews	80.73%
	n-grams, SentiWordNet scores and syntactic features [Dang et al. 2010]	Product reviews	84.28%
	Surface, syntactic, semantic *n*-grams with FRN [Abbasi et al. 2011]	Movie/Product reviews	90.7%
	Document Embedding [Le and Mikolov 2014]	IMDB reviews	90.58%
	CNNs followed by average pooling [Tang et al. 2015b]	IMDB/YELP reviews	59% (5-way)
	CNNs followed by GRNNs [Tang et al. 2015a]	IMDB/YELP reviews	63% (5-way)

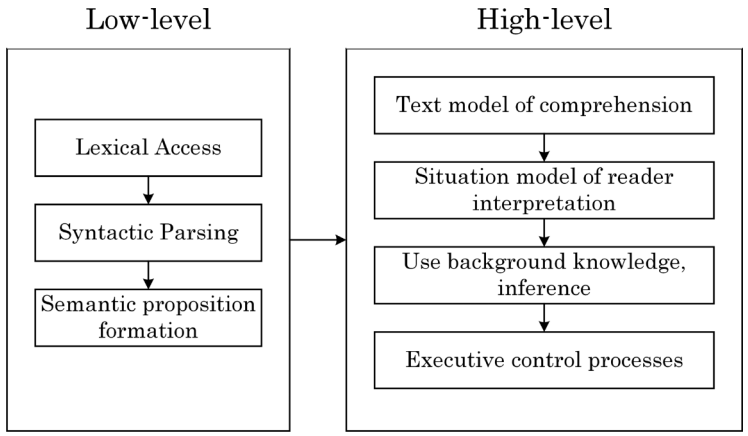

Fig. 1. The human reading process.

Table I illustrates the reported accuracies of the main approaches for DSC. These approaches cannot be directly compared one to another, as they were evaluated on different corpora; however, this table should be helpful in providing a perspective on what has been achieved so far.

3. PROPOSED MODEL

In this section, we describe the proposed human-based model for DSC. The idea is to augment shallow syntactic features with deeper semantics related to sentiment by automating aspects of the human reading process and deriving new features called "notions" as pairs of: {topic textual characteristic, sentiment score according to the human subject}. We first present the psychological foundation of the human reading process in Section 3.1, and then in Section 3.2 we motivate the need for the HRS model by identifying research gaps in existing state-of-the-art models.

3.1. Psychology Aspects of Human Reading Process and HRS Meta-Framework

Deeper understanding of text requires the integration of background information with the text being read. According to Grabe and Stoller [2013], the human reading process encompasses two levels: low-level and high-level processing, as shown in Figure 1.

At low-level processing, readers process the document at granular levels. They first perform lexical parsing to convert the sequence of characters into a sequence of tokens. The next step involves syntactic parsing using the grammatical rules by which sentences are constructed. Finally, readers develop a semantic proposition based on lexical and syntactic parsing to form sub-sentence structures called clauses, each representing a certain idea or concept.

At high-level processing, inference takes several steps within the human's cognitive process. First, readers use the purpose of reading with their background knowledge to decide which clauses represent the document. Clauses that do not fit the reading purpose are discarded and the remaining ones form the "*text model of comprehension*." Based on the readers' background knowledge and preconceived notions on the topic, the "*text model of comprehension*" is then transformed into the "*situation model of reader interpretation*" by assigning a sentiment label to each clause. Consequently, readers can infer the overall sentiment orientation of the document.

Inference is an important factor that reflects how humans reach conclusions through reading [Kurland 2000]. To infer a text's overall meaning, readers must have certain facts or evidence in their memory when reaching the end of the text. The retaining of facts is a part of the working memory (WM), which combines temporary storage (short-term memory) with information manipulation during reading comprehension to make sense of the text [Baddeley and Hitch 2010]. The working memory directly affects reading comprehension, where humans with low WM have difficulty in comprehension [Dewar 2012]. For example, by the time a child finishes spelling the first few words, he may forget what he intended to say. Similarly, a child would have troubles with reading comprehension; while he or she is working hard to decode written words, he or she may lose track of the overall purpose of the text.

The above-mentioned human reading steps provide the foundation to achieve automated machine reading and to derive the notions features for automated sentiment classification. The conforming frameworks must be able to accurately capture and model the low-level and high-level human reading processes shown in Figure 1. To use HRS with existing sentiment analysis methods, two steps are followed. First, a qualitative analysis of the method is performed against the human reading steps to identify potential gaps. Then, approaches to address the identified gaps are developed and integrated into that method. These approaches vary depending on its underlying learning scheme. At the high-level, addressing gaps in feature engineering-based methods can be done by proposing new features or by modifying the classification model. Addressing these gaps in deep learning-based methods can be done by modifying the neural network architecture or the network's raw input data.

3.2. Research Gaps with Respect to HRS

The automation of human reading is equivalent to machine reading (MR), which refers to the task of "understanding text" by automatically forming a coherent set of beliefs based on a textual corpus and a background theory [Etzioni et al. 2006]. Although humans' ability to grasp complicated nuances from text greatly surpasses that of a machine, MR still has several strengths; it is fast and can leverage statistics from large-scale corpora [Etzioni et al. 2006]. Different tools have been developed for machine reading such as KnowItAll [Etzioni et al. 2005], TextRunner [Banko et al. 2007], and Kylin [Wu and Weld 2007] for open-domain information extraction; Mulder [Kwok et al. 2001] for web-scale question answering; and HOLMES [Schoenmackers et al. 2008] for inference.

Here, we go deeper into the most relevant sentiment classification methods from the related work and provide a qualitative analysis with the aim of highlighting the HRS-specific gaps that need to be addressed. The objective is to evaluate how well each

Table II. Qualitative Analysis of Recent DSC Methods Against the Human Reading Aspects

	Syntactic Parsing	**Semantic proposition formation**	**Text model of comprehension**	**Situation model of interpretation**	**Inference & Memory**
Yessenalina et al. [2010]	N/A	N/A (classify based on unigrams)	Optimized solution to determine important sentences in a review	N/A	Classify reviews using unigrams in identified "important" sentences
Abbasi et al. [2011]	Use POS tag n-grams and word-POS tag n-grams	Represented by n-grams	FRN keeps only relevant features after reduction	N/A (capture general semantics, but not sentiment)	Train SVM using relevant n-grams (equally-important)
Tu et al. [2012]	Perform syntactic parsing to represent reviews' sentences	Identify sub-structures (clauses) from parse trees	Assume reviews must be represented by selected subjective clauses	N/A	Train SVM with subjective clauses represented using trees kernels (equally-weighted)
Le and Mikolov [2014]	Word-level syntactic properties captured via embedding	N/A (derive document vector using word vectors)	N/A (used all words to derive the document vector)	N/A (word embedding captures general semantics not sentiment)	Train Softmax using derived document vectors
Glorot et al. [2011]	Word-level syntactic properties captured via embedding	Derive document vector using 1- and 2-grams	N/A (used all words to derive document vector with stacked denoising autoencoders)	N/A	Train SVM using document vectors derived using SDA
Tang et al. [2015b]	Word-level syntactic properties captured via embedding	Use convolutional neural networks with multiple filters to capture n-grams semantic	N/A	N/A	Use document vectors (average sentence vectors) with author/product features to train Softmax
Tang et al. [2015a]	Word-level syntactic properties captured via embedding	Use convolutional neural networks (CNNs) with multiple filters to capture n-grams semantic	Weights are optimized in CNNs to select what n-grams will be used for modeling	N/A	Use GRNN to derive document vectors and learn weights that model the impact of each sentence on the overall document

method captures each of the different human reading steps mentioned in Section 3.1, namely (1) syntactic parsing (the use of the language grammar), (2) semantic proposition (the identification of constructs that match concepts or ideas), (3) text model of comprehension (the retaining of relevant ideas), (4) situation model of interpretation (the sentiment labeling of relevant ideas), and (5) inference and proper modeling of memory. The results of the qualitative analysis are summarized in Table II. The columns of the table reflect the different aspects of the human reading process, while the rows list the most relevant articles for DSC.

Table II provides a capture of gaps in previous works. For example, Yessenalina et al. [2010] proposed to predict document-level sentiment while jointly selecting key sentences in every document. This method clearly captures and optimizes the "*text model of comprehension*" but misses modeling the remaining aspects. The method proposed by Tu et al. [2012] represent documents by selected subjective clauses, which

is consistent with the "*semantic proposition formation*" and the "*text model of comprehension*." However, the model does not incorporate sentiment in these clauses, hence fails to capture the "*situation model of interpretation*." The method proposed by Le and Mikolov [2014] derives document vector representations that captures the semantics and context of all words in every document. This model incorporates information from all words and sentences, with equal considerations, into the document vector, and hence does not meet the "*text model of comprehension*" aspect, which assumes that some parts of the document are more important than others. Additionally, the resulting document representation does not reflect the sentiment content of its composite words and hence does not capture the "*situation model of interpretation*" aspect.

Table II shows that the different methods exhibit different gaps in comparison with HRS. This observation confirms that previous methods that were developed to improve DSC handle some, but not all, aspects of the human reading process. The proposed HRS provides a unifying meta-framework to address the human-reading gaps with any existing method. The design and structure of HRS depend on the underlying learning scheme. The details of adjusting different sentiment analysis methods to HRS are presented in the next section.

4. APPLICATIONS OF THE HRS META-FRAMEWORK

In this section, we describe the proposed HRS meta-framework to automate human reading with primary focus on understanding sentiment instead of the full semantics of the text. To demonstrate the effectiveness of HRS, we apply it to state-of-the-art methods representing today's schools of ML approaches. FRN [Abbasi et al. 2011] was selected as a representative of ML methods that use feature engineering followed by classification and achieved high accuracies among such methods. On the other hand, GRNN [Tang et al. 2015a] is chosen to represent the latest advances in deep learning approaches that derive inferencing from raw data without going through feature engineering, achieving highest performances on different benchmark datasets. It is worth mentioning that frameworks resulting from the HRS meta-framework must be consistent with the learning method being used. Consequently, HRS steps for feature engineering-based methods would differ from HRS steps for deep learning-based methods.

4.1. HRS Application to ML Approaches with Feature Engineering: FRN as Case Study

Abbasi et al. [2011] explored a wide range of surface, stylistic syntactic, and semantic features and proposed FRN to retain only the relevant and non-redundant ones. The algorithm ranks the features using semantic information about the subjectivity of each feature. It also uses a score that represents the discriminating power with respect to the different classes. Then, every feature is assigned a weight that is equal to the sum of both semantic and discriminating scores. These weights are then used by the subsumption and parallel relations to remove redundant or irrelevant features that do not convey any extra information to the classification process. The resulting feature subset is then used to train a SVM classification model.

First, we perform a macro-level analysis to identify gaps in FRN with respect to HRS. FRN uses a wide range of features and captures well the low-level processes through the NLP task of extracting shallow text features. To automate "*lexical and syntactic parsing*," FRN uses a syntactic base created through tokenization and POS tagging. However, it partially captures the "*semantic proposition formation*" by extracting *n*-grams instead of phrases. It also fails at modeling the high-level processes of "*semantic proposition formation*," "*situation model of interpretation*," and "*working memory*." In other words, FRN captures most of the low-level processes but misses the high-level processes.

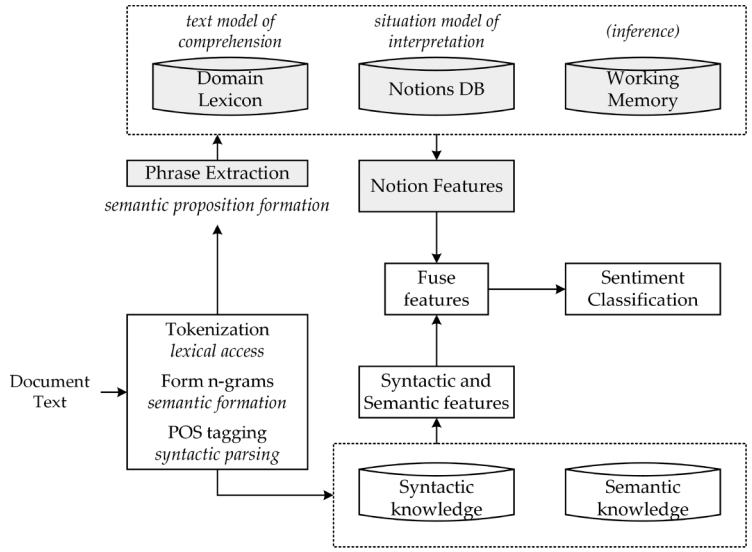

Fig. 2. Application of the HRS meta-framework to the FRN method.

Next, we address the identified gaps in FRN as shown in Figure 2, which depicts the HRS framework proposed to incorporate the missing HRS aspects highlighted in gray. Instead of using n-grams to model the "*semantic proposition formation*," we perform phrase extraction by splitting sentences according to the presence of coordinate and subordinate conjunctions and punctuation [Hopper 1986]. To address gaps in the high-level processes, we propose to extract the "notions" features. This stage of semantic analysis requires the development of (1) a domain lexicon that contains domain-specific keywords for modeling the "*text model of comprehension*," (2) a background knowledge database of notions that assigns sentiment scores to selected phrases for modeling the "*situation model of interpretation*," and (3) a weighting scheme that assigns a memory score to each notion based on its location in the document to emulate the impact of the "*working memory*." At last, inference is achieved through a classifier that is trained using the notions features and other syntactic and semantic features proposed by Abbasi et al. [2011].

Below, we describe the details of the resulting HRS framework that is proposed to address the human reading gaps in FRN. In particular, we describe the syntactic and semantic features proposed by Abbasi et al. [2011] to model the low-level human reading processes. We also describe the proposed approach to extract notions features to model the high-level aspects of the human reading process.

4.1.1. Syntactic and Semantic Features (For Low-Level Processing). The FRN method [Abbasi et al. 2011] relied on a rich set of syntactic and semantic n-gram features. We will refer to these features as the "FRN" features. The following is a brief description of these features.

—n-words and n-characters: Sequences of characters and words of different lengths n ranging from 1 to 3.
—n-POS tags: Sequences of POS tags of different lengths n from 1 to 3.
—n-POSwords: Sequences of words and POS tags of different lengths n from 1 to 3.
—n-legomena: Consist of normal n-word features, with the difference that words occurring only once are replaced with the word "HAPAX," and words occurring only twice in the corpus are replaced with the word "DIS."

—n-semantic: These are n-words where words that belong to some synonym group in WordNet are replaced by their group label. The label becomes a unique ID identifying the specific synonym group.
—Information Extraction Patterns: A set of syntactic templates that indicate subjective content (e.g., passive-verb: "was destroyed") [Riloff et al. 2006].

The size of this feature set is huge, reaching millions of features for moderate-size corpora. Hence, it is necessary to perform feature reduction in order to keep the relevant ones for later use in the HRS model. Many feature selection algorithms have been proposed, including the Entropy-weighted genetic algorithm (EWGA) [Abbasi et al. 2008], FRN [Abbasi et al. 2011], and feature ranking using different measures such as proportional difference of SentiWordNet scores [OKeefe and Koprinska 2009], information gain, χ^2, document frequency [Yang and Pedersen 1997] and standard deviation [Yousefpour et al. 2014]. We decided to use FRN, which achieved high performance when used to reduce the above-mentioned "FRN" features.

4.1.2. Notion Features and Background Knowledge. These features are proposed to capture the following human reading aspects: the "*semantic proposition formation*," the "*text model of comprehension*," and the "*situation model of interpretation*." In general, the human's preconceived notions enable a person to decide whether a text has positive, negative, or neutral sentiment. We define the equivalent of a human's preconceived notions on a topic as pairs of {topic textual characteristic, sentiment score according to the human subject}. For example, {*low battery consumption*, (+) score} and {*low resolution*, (−) score} represent positive and negative notions in someone's mind, respectively. To define the notions' textual characteristics, we assume that the smallest text unit that contains a notion is a phrase that is composed of adjectives, adverbs, nouns, and verbs. Phrases are identified in sentences based on the presence of known coordinate and subordinate conjunctions (and, but, for, etc.) as well as punctuation (commas, periods, etc.).

However, not all phrases can be considered as notions in a particular domain, as some could present ideas that are general or irrelevant in building the set of domain-specific notions. A domain lexicon is developed to evaluate the relevance of a phrase to become a notion. The domain lexicon contains nouns and verbs that appear frequently in a particular domain. Consequently, the notions database is formed by going through the documents and storing phrases that contain at least one domain word, be it a noun or verb. Then, each phrase (the notion's textual representation) needs to be associated with a sentiment score that indicates the degree of negativity or positivity expressed by the phrase. This score is automatically obtained for each phrase by calculating the difference in pointwise mutual information (PMI) between that phrase and the sentiment classes [Kiritchenko et al. 2014], as shown in Equation (1).

$$\text{Sentiment of phrase } p = \text{PMI}_{p,\,pos} - \text{PMI}_{p,\,neg} = \log_2 \frac{\text{freq}(p, pos) \times \text{freq}(neg)}{\text{freq}(p, neg) \times \text{freq}(pos)}, \quad (1)$$

where freq(p, pos) and freq(p, neg) are the occurrence frequency of phrase p in positive and negative documents, respectively. freq(pos) and freq(neg) are the count of positive and negative documents in the corpus, respectively. The sentiment scores are then normalized to fall in the range between $[+1, -1]$. The resulting background knowledge database consists of phrase-level notions related to a particular domain. When a new phrase is encountered, it is compared against the background knowledge database to match with the textual characteristics of the notions it contains. The matched notions then make up the document's notion features, which are used to infer the document-level sentiment.

It is worth mentioning that the background knowledge database is in some way similar to sentiment lexicons with the following distinctions. First, it consists of phrases while most existing sentiment lexicons consist of words or word n-grams. Second, its entries pertain to a particular domain rather than being generic, which allows it to model humans' notions on a particular topic. Third, it is automatically generated and does not involve manual annotation.

4.1.3. Notions Synonyms (For Improved Generalization). When classifying new text, the proposed HRS tries to recognize notions that are previously stored in the background knowledge database. For training, the matches help in grouping similar notions together in the notions database. For classification, the matches help in assessing the sentiment polarity by measuring similarity to a previously stored notion. However, different people may express the same idea through different words, hence a limited set of notions expressions would limit the matching process. For this reason, it is important to store the notions in a way that allows generalization of notions with either exact word matches or synonymous semantically related words.

WordNet [Fellbaum 1999] has been widely used by sentiment analysis systems as a rich source of semantic relations between words, hence providing generalization capabilities. For example, words were replaced by their hypernyms in WordNet as in Breck et al. [2007] or by their synonym sets (synset) labels as in Hassan et al. [2013]. Synsets were also used to create semantic categories by clustering words based on the number of common items in their synsets [Kim and Hovy 2004]. We propose to use WordNet synsets to create a database of notions' synonyms. The textual characteristics of existing notions are replaced by synonymous generalized text, which are combined with the sentiment scores from the original notions to create the notions' synonyms (SYN notions). For instance, notions with the following textual characteristics: "*passion movie*" and "*love movie*," are synonymous because *passion* and *love* belong to the same WordNet synset whose label is 40. Therefore, when creating the SYN notions, each of these words is replaced by its synset label; 40. As a result, the textual characteristics of both notions are unified into *"SYN40 movie."*

4.1.4. Human Working Memory (for Improved Inference). The goal of modeling the working memory is to bring the automation process closer to the way a human reader processes, analyzes, and draws conclusions from a document. As readers progress through text, they are more likely to remember later sentences, which are stored in short-term memory [Dewar 2012]. Consequently, we propose to assign each notion in a document a weight that models the impact of the human working memory. The weighting scheme assigns weights depending on the notion's position within the document; weights are low for notions that appear at the beginning and become higher for notions that appear at later parts. This weighting scheme is illustrated in Equation (2),

$$\text{working memory weight of notion } p = \sum_{i=1}^{n_p} \text{pos}_i \div \sum_{i=1}^{n} \text{pos}_i, \tag{2}$$

where n_p is the frequency of notion p in the document, pos_i is position of the i^{th} occurrence of notion p in the document, and n is the count of all notions in the document. Then, each notion is represented by the product of its sentiment score and its memory weight to reflect the fact that a reader's sentiment interpretation would be affected by what they read at the end more than what they read at the beginning.

Several previous works evaluated the importance of later sentences based on the hypothesis that authors tend to summarize their opinions at the end of the document [Becker and Aharonson 2010; Pang and Lee 2004], which confirms the necessity of the working memory for sentiment comprehension. According to Becker and

Aharonson [2010], by reading only the last sentence of a review, human readers were able to predict the reviews' sentiment polarity with performance that is comparable to the case when they read the whole review. Also, Pang and Lee [2004] showed that keeping the last N sentences, assuming their importance in summarizing the text sentiment, allowed us to exclude objective sentences.

The proposed WM scheme differs from previous work in the following aspects. First, it is applied to notions (phrases) instead of sentences. Second, it assigns gradual weights instead of a crisp decision of whether or not to include certain notions into the model, hence it takes into consideration all notions in the document with varying weights, which could remedy the issue that arises when classifying documents with no concluding statements. It is worth mentioning that the proposed working memory feature is expected to work best with a particular genre of text that contains a conclusion in the last sentences. Consumer reviews constitute a major portion of this genre.

4.2. HRS Application to Deep Learning Approaches: GRNN as Case Study

GRNN [Tang et al. 2015a] is a deep learning model that extracts sentiment using vector representations (embeddings) that capture the syntactic and semantic properties of the raw words [Mikolov et al. 2013]. GRNN follows a hierarchical architecture of three stages. First, input word vectors are used by the CNNs [LeCun and Bengio 1995] to derive sentence representations. Second, the sentence representations are used by the GRNN to derive the document representation. At last, the document representation is used to train a Softmax classifier. To apply HRS to GRNN, we follow the same strategy of identifying HRS-related gaps in GRNN and proposing approaches to address these gaps.

Based on the qualitative analysis shown in Table II, GRNN already models many aspects of the human reading process. Word embedding is responsible for the lexical access and syntactic parsing. CNNs use three convolutional filters to encode the semantics of n-grams, which is equivalent to the "*semantic proposition formation*" with phrases being limited to trigrams. The GRNN model used in the second stage of the hierarchy learns long- and short-term events by including a "forget" gate acting as a switch that erases or keeps the dependency between the current state and past events. Hence, the network is able to develop a model that acts like the human working memory, where readers develop their understanding of the text based on what they recently read and forget older history. The main gap in GRNN is the lack of modeling the "*situation model of interpretation*," where the intermediate phrase and sentence representations do not capture sentiment information. Another limitation is the inability to perform "*semantic proposition formation*" with phrases longer than trigrams due to the restriction in the CNNs filters size. As the latter issue does not represent a major obstacle towards incorporating HRS with GRNN, we focus on modeling the "*situation model of interpretation*" gap.

Figure 3 illustrates the GRNN architecture with and without HRS. We propose to address the "*situation model of interpretation*" gap by including word-level sentiment embedding to derive word vectors, where each vector captures the sentiment of its corresponding word. After obtaining the sentiment vectors, each word is represented by concatenating both the original embedding vector and the derived sentiment vector. The resulting representation includes information about syntactic, semantic, and sentiment aspects of the word. CNNs map this information to higher-level constituents when deriving phrase and sentence representations. As a result, these representations will include sentiment indicators, making them equivalent to notions.

The proposed approach for sentiment embedding is inspired by word embedding; a neural language model that encodes the context of each word into a dense, low-dimensional, and real-valued vector [Frome et al. 2013]. The idea is to train a network

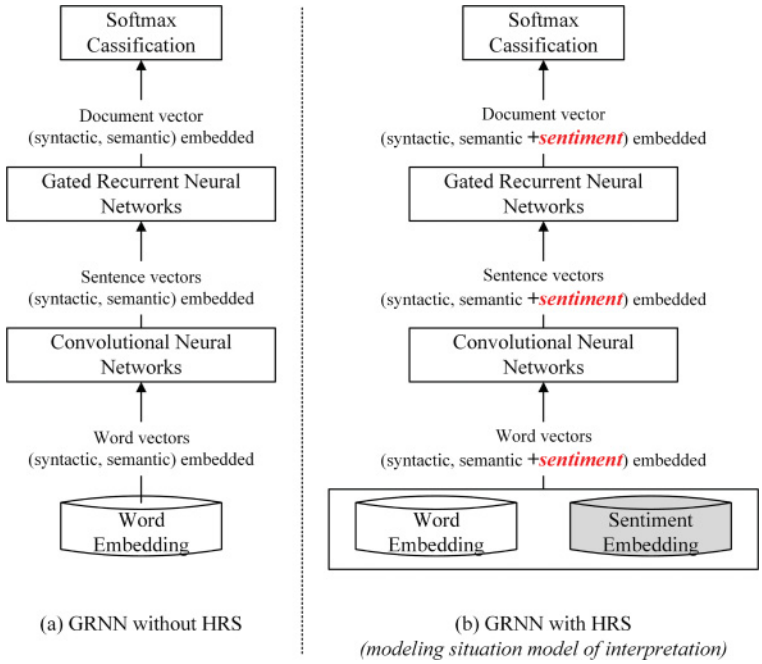

Fig. 3. The architecture of GRNN with and without HRS. (a) The original GRNN framework; (b) the modified GRNN framework with the modeling of the "*situation model of interpretation*."

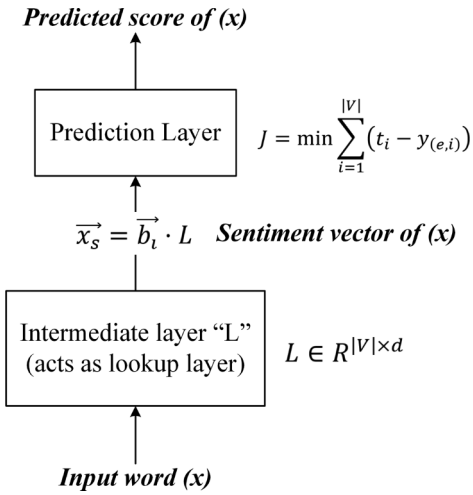

Fig. 4. The neural network architecture proposed to perform sentient embedding.

with the objective of learning the sentiment score of each word. Figure 4 illustrates the proposed neural network architecture for sentiment embedding.

The network parameters of both the prediction and the intermediate layers are optimized using gradient descent in back-propagation with respect to the objective function J that minimizes the error between the words' actual sentiment scores and

Table III. Characteristics of Datasets Used in Experiments

Corpus	Size	Domain	Classes	Distribution	splits (Tr, Dev, Test)
IMDB	2K reviews	movies	2	50 - 50%	70%, 10%, 20%
YELP	10K reviews	restaurants	5	9 - 9 - 14 - 33 - 35% (very neg to very pos)	80%, 10%, 10%

the network-generated scores, as shown in Equation (3),

$$J = \min \sum_{i=1}^{|V|} \left(t_i - y_{(e,i)}\right), \tag{3}$$

where $|V|$ is the size of the vocabulary, t_i and $y_{(e,i)}$ are the original and network-generated sentiment scores of the ith word, respectively. The objective function can also be formulated as maximizing the conditional probability of obtaining the desired word sentiment score t_i given the word input vector x_i.

The desired output of the proposed sentiment embedding are the weights of the intermediate layer L, which act as a lookup table $L \in \mathbb{R}^{|V| \times d}$, where d is the embedding size. This layer is equivalent to a fully connected network of $|V|$ input neurons, and d output neurons. Given an input word x, we can get its equivalent vector using a lookup operation as shown in Equation (4),

$$\overrightarrow{x_s} = \overrightarrow{b_i} \cdot L, \tag{4}$$

where $\overrightarrow{x_s} \in \mathbb{R}^{1 \times d}$ is the sentiment vector of word x, $\overrightarrow{b_i} \in \mathbb{R}^{1 \times |V|}$ is a one-hat vector that activates one input neuron in L.

The actual word sentiment scores can be obtained from available sentiment lexicons. In this article, we use SentiWordNet as the source of word-level sentiment scores for sentiment embedding. SentiWordNet assigns three sentiment scores for each word to express its degree of positivity, negativity, and neutrality. Equivalently, the prediction layer will output three scores, and the objective function will be obtained by minimizing the error over the three sentiment scores.

5. EXPERIMENTS AND RESULTS

In this section, we apply the HRS meta-framework to two state-of-the-art DSC methods, namely FRN and GRNN. We show that applying HRS to each method yields improvement even when performance is already high without HRS. The datasets used in the experiments are consistent with the choices made by the respective articles, enabling comparison of HRS with these methods as standalone. Subsection 5.1 describes the datasets, evaluation method, and metrics. Subsections 5.2 and 5.3 illustrate the impact of incorporating HRS with FRN and GRNN models, respectively.

5.1. Datasets and Evaluation

Two datasets are used for the quantitative analysis; the IMDB movie reviews and the YELP restaurant reviews. The IMDB dataset contains 2,000 reviews equally distributed across positive and negative sentiment classes [Pang et al. 2002]. The YELP dataset contains 10,000 restaurant reviews selected from the YELP dataset (2013 YELP dataset challenge).[1] Reviews in this dataset come with human-based ratings on a scale of 1 to 5. Table III highlights the characteristics of these datasets.

For evaluation, the IMDB dataset was randomly divided into a train set (70%), a development set (10%), and a test set (20%). As for YELP, the splits had the sizes

[1]www.yelp.com/dataset_challenge.

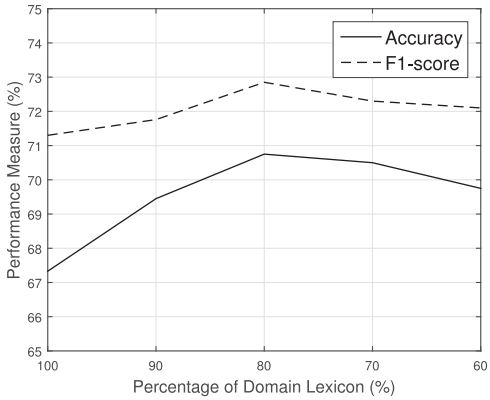

Fig. 5. Evaluating different sizes of the domain lexicon.

80%-10%-10% for train, development, and test, respectively. We used the train and test sets to tune the classifier's parameters. For final evaluation, the model is derived using the train and test sets with the parameters that achieved best performance in tuning. Results are reported in terms of accuracy and F1-score on the test set.

5.2. Evaluating FRN with HRS

As previously discussed, FRN does not model each of the "*semantic proposition formation*," the "*situation model of interpretation*," and the "*working memory*." To model these aspects with FRN, the HRS steps consist of (1) extracting phrases, (2) retaining domain-relevant phrases using domain lexicon, and (3) assigning each phrase a sentiment score and a working memory weight. Notion features are then fused with the FRN features to train a nonlinear SVM with radial basis function for sentiment classification.

5.2.1. Evaluating Text Model of Comprehension. To evaluate the text model of comprehension, we use different portions of the domain lexicon and study their effect on overall performance. The domain lexicon contains frequent nouns and verbs that pertain to the domain being discussed. To determine the frequency thresholds to be used to develop the domain lexicon, we performed a set of tuning experiments by training an SVM classifier using domain-relevant phrases extracted using different thresholds determined as follows. After removing stopwords, nouns (or verbs) are ranked based on their frequency. Then, the threshold is identified as the one that allows us to retain the top frequent nouns (or verbs) constituting $X\%$ of the total count, where X is varied between 60% and 100%. Both the domain lexicon entries and phrases are extracted from the training set to avoid over-fitting.

Results in Figure 5 show that using a domain lexicon whose entries constitute 80% of the total count of all nouns and verbs yields the highest performance as measured in both accuracy and F1-score. This observation confirms the concept of "*text model of comprehension*" in the human reading framework, which states that only domain-relevant phrases should be used. It can be observed that using $X = 100\%$ does not yield the best performance given that many phrases can be domain irrelevant. Also, reducing X down to 60% decreases performance as many domain-relevant phrases are excluded from the model. These results are consistent with the classic bias-variance tradeoffs in prediction problemsEighty percent gives the best tradeoff between using enough of the data to improve performance but not too much to avoid over-fitting.

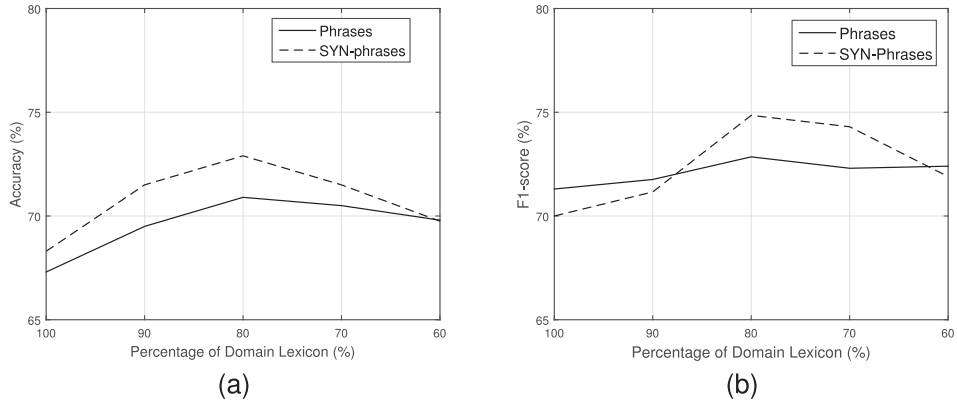

Fig. 6. Evaluating the impact of synonymy when introducing SYN-phrases vs. phrases under different sizes of the domain lexicon. Panel (a) shows the accuracies, while (b) shows the F1 scores.

5.2.2. Evaluation of Synonyms. To evaluate the impact of synonyms, two SVM classifiers are trained separately with either phrases or SYN-phrases that are extracted from the training set. This experiment is repeated for different sizes of the domain lexicon, similarly to the previous experiments. Results in Figure 6 show that using SYN-phrases yields better results, in both accuracy and F1, compared to using normal phrases. This is due to the improvement in generalization that takes place when synonymous phrases sharing similar meanings are collapsed together. It can also be observed that using a domain lexicon whose entries constitute 80% of the total count of nouns and verbs in the corpus yields highest results regardless of the type of phrases being used. The 80% threshold provides results consistent with the previous experiment and will be used later when evaluating FRN with the full HRS framework.

5.2.3. Comparison between HRS with FRN Versus Standalone FRN. We evaluate the FRN method with and without the proposed HRS. First, we extract the FRN features from the training set. We use kfNgram [Fletcher 2002] to extract the n-word and n-character features. We also use it with the Stanford tagger [Toutanova et al. 2003] to extract the n-POS features. The n-legomena features are extracted similarly to the n-words but after replacing all once- and twice-occurring words by "HAPAX" and "DIS," respectively. Similarly, n-semantic features are extracted after replacing each word by its corresponding synset label. At last, the Information Extraction Patterns are extracted using the Sundance package [Riloff et al. 2006]. For all n-gram features, the value of n varies between 1 and 3.

Second, we develop the proposed notions features as follows. After developing the domain lexicon with size of 80%, and selecting SYN-phrases to represent the textual characteristics of the notions, we proceed to develop the notions database by assigning a sentiment score to each SYN-phrase. The sentiment score of a SYN-phrase is defined as the difference in PMI scores calculated between that phrase and each of the sentiment classes, as illustrated in Equation (1). A score ≥ 0 implies that the notion has a positive sentiment, otherwise it is negative. Once the notions database is developed, each notion in a review is assigned a working memory weight that is based on its location in that review, as shown in Equation (2).

To reproduce the FRN experiments, the FRN algorithm[2] is applied to reduce the FRN features. The algorithm outputs several reduced feature sets that are then used to

[2]Source code of FRN was provided by Abbasi et al. [2011].

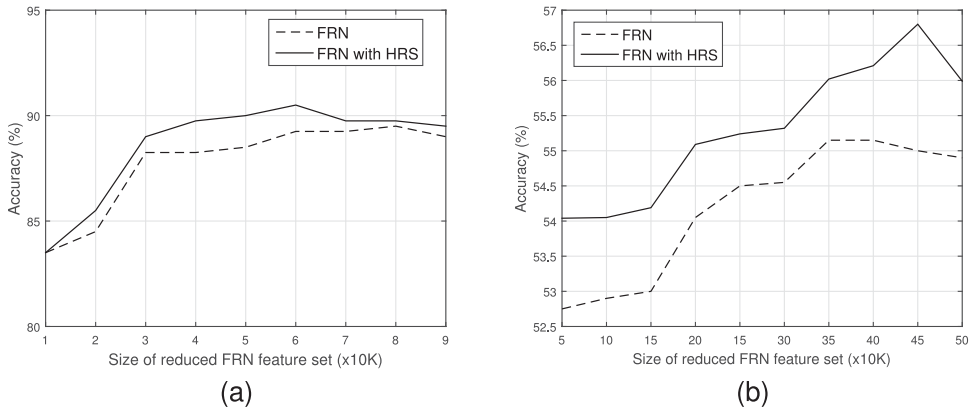

Fig. 7. Evaluating the FRN method with and without HRS under different sizes of reduced features. (a) Results on the IMDB corpus. (b) Results on the YELP corpus.

train the SVM model. For the IMDB corpus, the size of the reduced feature sets ranged between 10K and 100K features. For the YELP corpus, the sizes ranged between 50K and 500K since the corpus is bigger and contains many more features. To evaluate FRN with the HRS framework, we fuse the notions features with each of the different sets of reduced FRN features. Each notion is represented by the product of its sentiment score and its working memory weight in every document. The classification results in Figure 7 show that the proposed HRS framework introduces consistent improvements in accuracy on both datasets. On IMDB, and despite the high accuracy with FRN standalone close to 90%, HRS pushed the accuracy even higher with an average of 1%. Similarly on YELP, HRS improved the accuracy by 1.07% on average. It is worth mentioning that these performances can be further improved by optimizing each of the added HRS steps.

5.3. Evaluating GRNN with HRS

In this subsection, we evaluate the impact of applying the HRS framework to GRNN. As described in Section 4.2, GRNN does not model the "*situation model of interpretation*." To address this gap with HRS, we proposed a sentiment embedding technique to derive word-level vectors that encode sentiment information.

In the original GRNN model, words are represented with 200-dimensional embedding vectors derived using word2vec [Mikolov et al. 2013]. In the proposed setup with HRS, each word will be represented using the concatenation of two 100-dimensional embedding vectors. The first part is derived using word2vec to capture syntactic and semantic properties of the word, and the other part is derived using the proposed sentiment embedding to encode the word's sentiment. Both vectors are obtained using the training set to avoid over-fitting.

We use SentiWordnet as the source of word-level sentiment information to train the sentiment vectors. To derive sentiment vectors for as many words as possible, the lexicon is extended by including stopwords, punctuation, digits, and out-of-vocabulary (OOV) words. Stopwords, punctuation, and digits are assigned an objectivity score equal to 1 and positivity/negativity scores equal to 0. On the other hand, each OOV word is assigned scores that are equal to the average scores of all SentiWordNet words that co-exist with it in the same sentence.

After tuning the GRNN parameters using the training and the development sets, as done in Tang et al. [2015a], each model is evaluated on the test set. To minimize

Table IV. Results Obtained on Both Datasets (YELP and IMDB) using GRNN With and Without HRS

Corpus	Approach	Accuracy	F1-score					
			very neg.	negative	neutral	positive	very pos.	avg.
YELP	GRNN	**57.70**	65.53	36.70	36.00	71.07	30.57	**48.0**
	GRNN+HRS	**60.17**	67.91	39.22	39.31	73.19	34.88	**50.9**
IMDB	GRNN	**90.81**	–	90.88	–	90.78	–	**90.8**
	GRNN+HRS	**92.00**	–	92.25	–	91.19	–	**92.1**

Table V. Summary of All Results Obtained by Applying HRS to FRN and GRNN Approaches on the IMDB Corpus (Five Classes) and the YELP Corpus (Five Classes)

Approach	IMDB corpus		YELP corpus	
	Accuracy	Average F1	Accuracy	Average F1
FRN	87.9	87.8	54.2	40.9
FRN+HRS	88.7	88.6	55.3	41.6
GRNN	90.8	90.8	57.7	48.0
GRNN+HRS	92.0	92.1	60.2	50.9

bias in the results, experiments are repeated 50 times with different random choices of initialization weights. Table IV shows accuracies and class-level F1 scores averaged over the 50 rounds.

The results illustrated in Table IV show that applying HRS to GRNN results in performance improvement on both datasets. On YELP, adding HRS introduced 4.74% relative improvement in accuracy and 5.8% relative improvement in average F1 score. These are significant improvements given the challenge of five-way sentiment classification. On IMDB, adding HRS introduced 1.31% relative improvement in accuracy and 1.43% relative improvement in average F1 score.

Table V summarizes the results for comparison of the HRS success with FRN versus GRNN to evaluate the impact of the HRS meta-framework on ML approaches based on feature engineering versus methods based on deep learning. For FRN-related experiments, we report the average performance over different sizes of reduced feature sets. For GRNN-related experiments, we report the average performance across different rounds with different choices of weight initialization. Results in Table V show that the HRS frameworks resulting from each method yield improvement compared to the methods without HRS. It can also be observed that HRS produced more improvement to GRNN in comparison to FRN. One reason for this observation is that we relied on feature engineering with some heuristics, such as notions sentiment scoring and memory weighting scheme, to address the HRS-related gaps in FRN. On the other hand, in GRNN, optimization formulations were used to embed sentiment information and assign memory weights. It is important to note that, even with sub-optimal approaches to address HRS gaps in FRN, the resulting framework achieved a better performance.

6. CONCLUSION

We presented a novel sentiment analysis meta-framework inspired from the humans' natural process of reading and inferring sentiment from text. This process involves low-level and high-level processing and uses background knowledge to infer semantics in general and sentiments in specific. In the process, humans use preconceived ideas to deduce their conclusions. The proposed human reading for sentiment, called HRS, provides a meta-framework for identifying gaps in existing approaches and then provides improvements to these approaches. HRS was described for document-level sentiment classification as the process of automating low-level and high-level human reading processes. We showed how to apply HRS to approaches that rely on feature

engineering and to other methods that rely on deep learning. For feature engineering-based methods, we proposed to develop new "notions" features to model aspects of the human reading process. On the other hand, we proposed to modify the neural network architecture in deep learning-based method.

In particular, for FRN, we proposed to represent the preconceived ideas by pairs of {phrase elements of nouns and verbs, sentiment score} and called these combinations "notions." Furthermore, we proposed to automate two additional aspects of the human reading. The first mimics the human working memory, which reflects the reader's short span of attention. The second represents the human's ability to relate words with close meanings and to group similar synonyms into individual synonym groups. For GRNN, we proposed to model the notions by producing embedded representations of sentiments at multiple levels of the GRNN hierarchy. Experiments showed the performance improvements when applying HRS to both state-of-the-art methods highlighting its ability to improve methods with already high accuracy.

In the future, improvements can be made by optimizing each step in the human reading process and by optimizing the integration for the full sequence of HRS steps.

ACKNOWLEDGMENTS

This work was made possible by NPRP 6-716-1-138 grant from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inform. Syst.* 26, 3 (2008), 12.

Ahmed Abbasi, Stephen France, Zhu Zhang, and Hsinchun Chen. 2011. Selecting attributes for sentiment classification using feature relation networks. *IEEE Trans. Knowl. Data Eng.* 23, 3 (2011), 447–462.

Alan Baddeley and Graham Hitch. 2010. Working memory. (2010). http://www.scholarpedia.org/article/Working_memory.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *IJCAI*, Vol. 7. 2670–2676.

Israela Becker and Vered Aharonson. 2010. Last but definitely not least: On the role of the last sentence in automatic polarity-classification. In *Proceedings of the acL 2010 Conference Short Papers*. Association for Computational Linguistics, 331–335.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI*, Vol. 7. 2683–2688.

Hsinchun Chen and David Zimbra. 2010. AI and opinion mining. *IEEE Intell. Syst.* 25, 3 (2010), 74–80.

Yan Dang, Yulei Zhang, and Hsinchun Chen. 2010. A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intell. Syst.* 25, 4 (2010), 46–53.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*. ACM, 519–528.

Gwen Dewar. 2012. Parenting for the science-minded. (2012). http://www.parentingscience.com/working-memory.html

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 231–240.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, Vol. 6. 417–422.

Oren Etzioni, Michele Banko, and Michael J. Cafarella. 2006. Machine reading. In *AAAI*, Vol. 6. 1517–1519.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.* 165, 1 (2005), 91–134.

Noura Farra, Elie Challita, Rawad Abou Assi, and Hazem Hajj. 2010. Sentence-level and document-level sentiment mining for arabic texts. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 1114–1119.

Christiane Fellbaum. 1999. *WordNet*. Wiley Online Library.

W. Fletcher. 2002. KfNgram. Retrieved July 29 (2002), 2009.

Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, and others. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*. 2121–2129.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 513–520.

William Grabe and Fredricka L. Stoller. 2013. *Teaching and Researching: Reading*. Routledge.

Ammar Hassan, Ahmed Abbasi, and Daniel Zeng. 2013. Twitter sentiment analysis: A bootstrap ensemble framework. In *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 357–364.

Jim Hendler. 2013. Broad data: Exploring the emerging web of data. *Big Data* 1, 1 (2013), 18–20.

Roula Hobeica, Hazem Hajj, and Wassim El Hajj. 2011. Machine reading for notion-based sentiment mining. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 75–80.

Lindsay Hoffman. 2013. Reflecting on Twitter and Its Implications for Elections and Democracy. (2013). http://www.huffingtonpost.com/lindsay-hoffman/twitter-elections_b_2568989.html.

Vincent Foster Hopper. 1986. *1001 Pitfalls in English Grammar*. Barron's Educational Series.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 168–177.

Ioannis Manoussos Katakis, Iraklis Varlamis, and George Tsatsaronis. 2014. PYTHIA: Employing lexical and semantic features for sentiment analysis. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 448–451.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, 1367.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *J. Artif. Intell. Res.* (2014), 723–762.

Daniel J. Kurland. 2000. Inference: The Process. (2000). www.criticalreading.com/inference_process.htm.

Cody Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the web. *ACM Trans. Inform. Syst.* 19, 3 (2001), 242–262.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053* (2014).

Yann LeCun and Yoshua Bengio. 1995. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks* 3361, 10 (1995).

Chenghua Lin, Yulan He, Richard Everson, and Stefan Ruger. 2012. Weakly supervised joint sentiment-topic detection from text. *IEEE Trans. Knowl. Data Eng.* 24, 6 (2012), 1134–1145.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining Text Data*. Springer, 415–463.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 2014. TeamX: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. *SemEval 2014* (2014), 628.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013a. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013)*. Atlanta, Georgia, USA.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013b. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* (2013).

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 26–34.

Tim OKeefe and Irena Koprinska. 2009. Feature selection and weighting methods in sentiment analysis. *ADCS 2009* (2009), 67.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 271.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*. Association for Computational Linguistics, 79–86.

Rawkes. 2011. The moment Twitter lost Steve Jobs. (2011). http://rawkes.com/articles/the-moment-twitter-lost-steve-jobs

Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. 2006. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 440–448.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. *Proc. SemEval* (2014).

Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld. 2008. Scaling textual inference to the web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 79–88.

Richard Socher, Cliff C. Lin, Chris Manning, and Andrew Y. Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 129–136.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Citeseer, 1631–1642.

Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1422–1432.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *Proc. ACL*.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014. Coooolll: A deep learning system for twitter sentiment classification. *SemEval 2014* (2014), 208.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 173–180.

Zhaopeng Tu, Yifan He, Jennifer Foster, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012. Identifying high-impact sub-structures for convolution kernels in document-level sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 338–343.

Peter D. Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 417–424.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. ACM, 625–631.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resour. Eval.* 39, 2–3 (2005), 165–210.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. ACM, 41–50.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *ICML*, Vol. 97. 412–420.

Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1046–1056.

Alireza Yousefpour, Roliana Ibrahim, and Haza Nuzly Abdull Hamed. 2014. A novel feature reduction method in sentiment analysis. *Int. J. Innovat. Comput.* 4, 1 (2014).

Xiaoui Yu, Yang Liu, Xiangi Huang, and Aijun An. 2012. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Trans. Knowl. Data Eng.* 24, 4 (2012), 720–734.

Received August 2015; revised April 2016; accepted May 2016