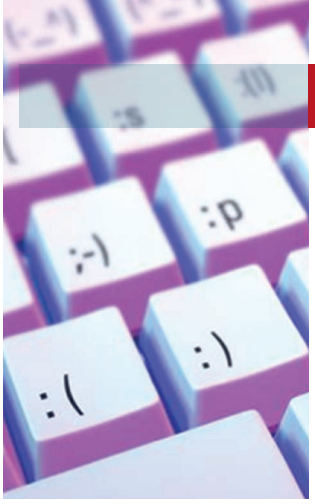

Editor: **Erik Cambria,** Nanyang Technological University, Singapore, cambria@ntu.edu.sg

Sentiment Analysis Is a Big Suitcase

Erik Cambria and Soujanya Poria, *Nanyang Technological University*
Alexander Gelbukh, *Instituto Politécnico Nacional*
Mike Thelwall, *University of Wolverhampton*

With the recent development of deep learning, research in artificial intelligence (AI) has gained new vigor and prominence. Machine learning, however, suffers from three big issues, namely,

- Dependency: it requires (a lot of) training data and is domain-dependent.
- Consistency: different training or tweaking leads to different results.
- Transparency: the reasoning process is uninterpretable (blackbox algorithms).

In the context of natural language processing (NLP), these issues are particularly crucial because, unlike in other fields, they prevent AI from achieving human-like performance. To this end, AI researchers need to bridge the gap between statistical NLP and many other disciplines that are necessary for understanding human language, such as linguistics, commonsense reasoning, and affective computing. They will have to develop an approach to NLP that is both top-down and bottom-up: top-down to leverage symbolic models such as semantic networks and conceptual dependency representations to encode meaning; bottom-up to use subsymbolic methods such as deep neural networks and multiple kernel learning to infer syntactic patterns from data.

Coupling symbolic and subsymbolic AI is key for stepping forward in the path from NLP to natural language understanding. Relying solely on machine learning, in fact, is simply useful to make a "good guess" based on past experience, because subsymbolic methods only encode correlation and their decision-making process is merely probabilistic. Natural language understanding, however, requires much more. To use Noam Chomsky's words, "You do not get discoveries in the sciences by taking huge amounts of data, throwing them into a computer and doing statistical analysis of them: that's not the way you understand things, you have to have theoretical insights."

It is necessary to take a holistic approach to sentiment analysis by handling the many subproblems involved in extracting meaning and polarity from text. Although most works approach it as a simple categorization problem, sentiment analysis is actually a suitcase research problem that requires tackling many NLP tasks (see Figure 1). As Marvin Minsky would say, the expression "sentiment analysis" itself is a big suitcase (like many others related to affective computing, such as emotion recognition or opinion mining) that all of us use to encapsulate our jumbled idea about how our minds convey emotions and opinions through natural language.

We address the composite nature of the problem via a three-layer structure inspired by the jumping NLP curves' paradigm (see Figure 2).[1] In particular, we argue that there are (at least) 15 NLP problems that need to be solved to achieve human-like performance in sentiment analysis. Such NLP problems are organized into three layers: syntactics, semantics, and pragmatics. The sequence of the different modules is just indicative and might require reordering depending on the specific data or domain being processed. For example, named entity recognition (NER) might have to be performed before lemmatization as some named entities would not be recognizable after lemmatization

1541-1672/17/$33.00 © 2017 IEEE
Published by the IEEE Computer Society

(such as "Guns N' Roses" versus "gun n rose").

Syntactics Layer

The syntactics layer aims to preprocess text so that informal text is reduced to plain English, inflected forms of verbs and nouns are normalized, and basic sentence structure is made explicit.

Microtext Normalization

The proliferation of social web technologies and the increasing use of computer-mediated communication has resulted in a new form of informal written text, termed *microtext*, which is characterized by relaxed spelling and reliance on abbreviations, acronyms, and emoticons.

This is partly a consequence of Zipf's law, or principle of least effort (for which people tend to minimize energy cost at both individual and collective levels when communicating with one another), and it poses new challenges for NLP tools, which are usually designed for well-written text.

The first step in tackling the challenge of developing algorithms to correct the nonstandard vocabulary found in microtexts is to realize that the number of different spelling variations might be massive, but they follow a small number of simple basic strategies, such as abbreviation and phonetic substitution. Although most of the literature on microtext normalization exploits supervised learning, unsupervised approaches have recently gained increasing popularity,[2] as microtext evolves too quickly to construct a comprehensive set of training data.

Sentence Boundary Disambiguation

In any document-level NLP task, sentence boundary disambiguation (SBD) is an important subtask that involves deconstructing text into sentences. SBD is particularly tricky when sentence boundary identifiers are not clearly defined, such as in the presence of emoticons. In fact, the accuracy of SBD can be improved after performing microtext normalization.

Early works employed decision trees to identify whether the presence of a full stop in text indicates the boundary of a sentence. Later, Jeffrey Reynar and Adwait Ratnaparkhi employed maximum entropy learning to create a sentence segmentation classifier that considered sentence boundary detection as a boundary disambiguation task, where every token containing "!", ".", or "?" was a potential sentence boundary.[3]

Part-of-Speech Tagging

POS tagging is a fundamental NLP task that labels each word by its part of speech, such as adjective, verb, and noun.

Most of the existing works consider POS tagging as sequence labeling task. The WSJ-PTB (the *Wall Street Journal* part of the Penn Treebank Dataset) corpus contains 1.17 million tokens and has been widely used for developing and evaluating POS tagging systems.

One characteristic of the POS tagging problem is strong dependency between adjacent tags. With a simple left-to-right tagging scheme, it is possible to model dependencies between adjacent tags only by feature engineering.

In an effort to reduce feature engineering, Zhiheng Huang and his colleagues concatenated word embeddings and manually designed word-level features and employed a bidirectional long short-term memory (LSTM) network to model wider context arbitrarily.[4]

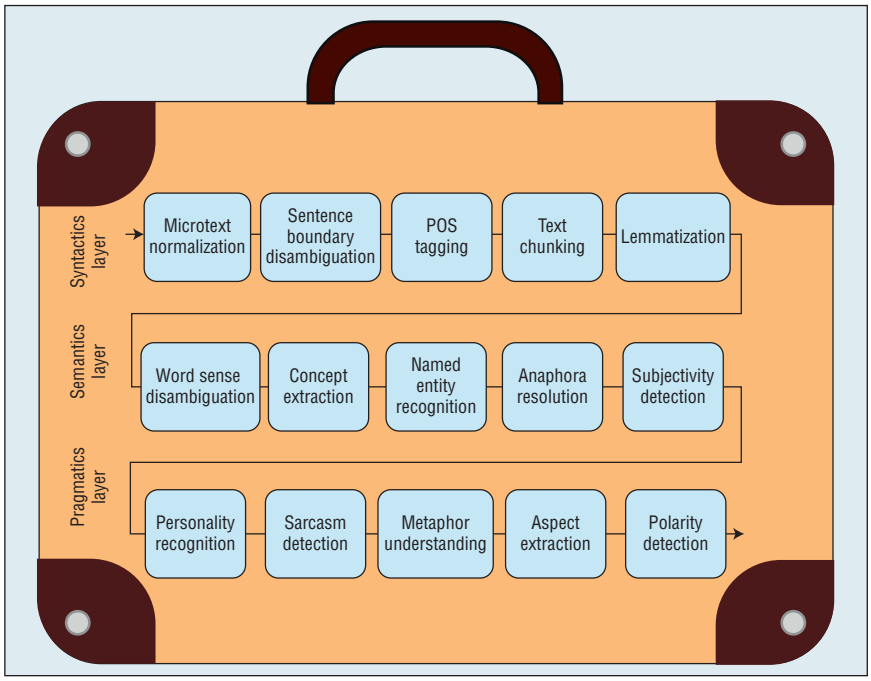

Figure 1. Sentiment analysis's big suitcase of natural language processing (NLP) problems. Sentiment analysis has long been mistaken for the task of polarity detection. This, however, is just one of the many NLP problems that needs to be solved to achieve human-like performance in sentiment analysis.

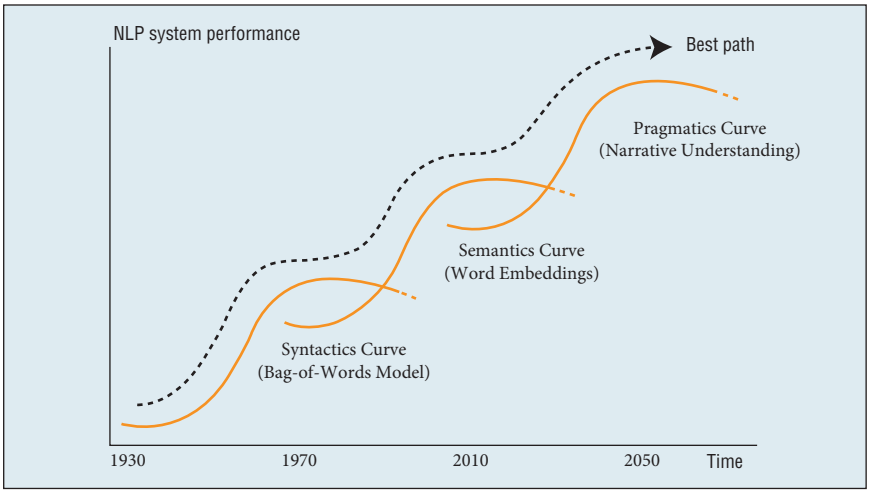

Figure 2. Jumping NLP curves. Borrowed from the field of business management and marketing prediction, this paradigm reinterprets the evolution of NLP research as the intersection of three overlapping curves, which will eventually lead NLP research to evolve into natural language understanding.

Text Chunking

Text chunking, also known as shallow parsing, follows POS tagging and adds more structure to the sentence. The result is a grouping of the words in simple syntactic substructures (chunks), such as noun groups or verb argument structure. Unlike full syntactic parsing, which implies building a deeply nested structure, text chunking splits sentences into nonoverlapping groups of words that represent a syntactic unit, without attempting to identify their internal structure or identify their relationships. An example of a chunked sentence is: "[NP *European Commission*] [VP *will probably launch*] [NP *a legal case*] [PP *against*] [NP *Poland*]."

Typically, a chunker uses many more superficial text analysis techniques than a full syntactic parser, and thus is much simpler, much more robust, and less resource-intensive. In particular, the chunks are usually unambiguous and do not depend on the choice of a syntactic formalism. Grouping words into chunks permits further identification of important basic relations between words, such as concept names and basic syntactic roles. Common approaches to chunking include rule-based and machine learning methods, such as transformation-based learning.[5]

Lemmatization

Lemmatization is the process of converting a given word into a base form, that is, a morphologically correct root form. This is useful for detecting a concept like `eat_burger` from all its possible inflected forms, for example, `ate_burger`, `eat_burgers`, `eating_burgers`, `eaten_burger`. Unlike stemming, lemmatization is meaning-preserving since it does not chunk away suffixes, regardless of semantics, and it preserves the word's POS tag. While a lemmatizer would reduce plurals to singular and inflected forms of verbs to their infinitive form, a stemmer would brutally remove morphological and inflexional endings from words. For example, it would reduce the words democrats (noun), democratic (adjective), and democratize (verb) to the same root (democrat). A recent approach to lemmatization is a deep learning method by Mike Kestemont and his colleagues,[6] who applied temporal convolutions to model the orthography of input words at the character level and used distributional word embeddings to represent the lexical context surrounding the input words.

Semantics Layer

The semantics layer aims to deconstruct the normalized text obtained from the syntactics layer into concepts, resolve references (that is, named entities and anaphora), and filter out neutral content from the input to improve sentiment classification accuracy.

Word Sense Disambiguation

A word can have multiple meanings depending on its context. WSD recognizes which sense of an ambiguous word is used in the input sentence and is key to improving the accuracy of concept extraction. WSD can benefit from POS tagging for disambiguating some word senses, for example, `fine` as noun (penalty) or `fine` as an adjective (good). Sometimes, however, POS tags are not enough; for example, the word `train` can be disambiguated as a verb but not as a noun (that is, `train` as in "high-speed train" versus `train` as in "train of elephants").

Yoong Keok Lee and Hwee Tou Ng evaluated a variety of knowledge sources and supervised learning algorithms for WSD on SENSEVAL-1 and SENSEVAL-2 data.[7] Their training data consisted of sentences having ambiguous words with their sense tagged manually. The knowledge sources used were POS tags of neighboring words, unigrams in the surrounding context, local collocations, and syntactic relations.

Concept Extraction

Deconstructing text into concepts is key for a semantic-aware analysis of text. Concepts can be either single words or multiword expressions, the latter being semantic atoms that should never be broken down into single words. The concept `pain_killer`, for example, should not be split into `pain` and `killer`, which are two words with completely different semantics (and polarity).

Current approaches can be classified into two main categories: statistical methods and linguistic rules. The former usually leverages term frequency and word location to calculate term weighting, the latter often uses POS tagging and text chunking to extract meaningful n-grams. A recent approach leveraged conceptual primitives automatically discovered through hierarchical clustering and dimensionality reduction.[8] Verb concepts such as `eat`, `slurp`, and `munch` were represented by their conceptual primitive `INGEST`, and noun concepts like `pasta`, `noodles`, and `toast` were replaced with their ontological parent `FOOD`. This way, any verb-noun combination between such concepts (for example, `eat_pasta`, `slurp_noodles`, or `munch_toast`), would be generalized and extracted as `INGEST_FOOD`.

Named Entity Recognition

NER is a subtask of information extraction that aims to locate and classify named entities into predefined categories. CoNLL 2003 is the standard English dataset for NER and concentrates on four types of named entities: persons, locations, organizations, and miscellaneous. NER is important for many other NLP tasks involved in sentiment analysis, especially anaphora resolution and aspect extraction.

NER is often broken down into two distinct problems: detection of names, typically simplified to a segmentation problem, and classification of names by the type of entity they refer to. Although early approaches mostly leveraged a domain-specific lexicon, recent works mostly employ deep learning on a training set.

Yukun Ma and his colleagues presented a label embedding method that incorporates prototypical and hierarchical information to learn pretrained embeddings and adapted a zero-shot learning framework that can predict both seen and previously unseen entity types.[9]

Deconstructing text into concepts is key for a semantic-aware analysis of text. Concepts can be either single words or multiword expressions, the latter being semantic atoms that should never be broken down into single words.

NER includes other NLP subtasks, such as temporal tagging, where general heuristic rules are usually employed to recognize time expressions in text[10] and which is sometimes more difficult than NER itself because, although standard named entities are usually expressed using formal language, temporal entities are subject to short expressions, small vocabulary, recurrence, and similar syntactic behaviors.

Anaphora Resolution

Anaphora can be defined as the presupposition that points back to a previous item. The pointing back reference is called an *anaphor* and the entity to which it refers is its *antecedent*. The process of determining the antecedent of an anaphor is called *anaphora resolution*, which is still an open NLP challenge that needs to be tackled in many domains, including machine translation, summarization, and question-answering. In the context of sentiment analysis, this task is key to resolving pronouns in text before subjectivity detection or aspect extraction can be applied.

The most widespread types of anaphora are pronominal anaphora, which is realized by anaphoric pronouns; adjectival anaphora, realized by anaphoric possessive adjectives; and one-anaphora, the anaphoric expression is realized by a "one" noun phrase. When resolving anaphora, some constraints must be respected: number agreement (to distinguish between singular and plural references); gender agreement (to distinguish between male, female, and neutral genders); and semantic consistency (it is assumed that both the antecedent clause and the one containing the anaphora are semantically consistent). Grammatical, syntactic, or pragmatic rules have been widely used in the literature to identify the antecedent of an anaphor. Whereas early works focused on the use of parse trees, discourse models, POS tagging, and lexical databases, more recent approaches have leveraged genetic algorithms.[11]

Subjectivity Detection

Subjectivity detection is a NLP task that aims to remove "factual" or "neutral" content (that is, objective text that does not contain any opinion) from online reviews. This

preprocessing step is crucial to increase the accuracy of sentiment analysis systems, which are usually optimized for the binary classification task of distinguishing between positive and negative content.

Previous methods used well-established general subjectivity clues to generate training data from unannotated text. In addition, features such as pronouns, modals, adjectives, cardinal numbers, and adverbs have shown to be effective in subjectivity classification. Some existing resources contain lists of subjective words, and some empirical methods in NLP have automatically identified adjectives, verbs, and n-grams that are statistically associated with subjective language. However, several subjective words such as "unseemingly" occur infrequently, and consequently a large training dataset is necessary to build a broad and comprehensive subjectivity detection system.

Recently, Iti Chaturvedi and her colleagues proposed a novel framework that exploits the features of both Bayesian networks and fuzzy recurrent neural networks for filtering out neutral content in a time- and resource-effective manner.[12]

Pragmatics Layer

The pragmatics layer aims to extract meaning from both sentence structure and semantics obtained from syntactics and semantics layers, respectively. After performing some kind of user profiling (personality and sarcasm detection), the pragmatics layer interprets metaphors (if any) and extracts opinion targets and the polarity associated with each of them.

Personality Recognition

Personality is a combination of an individual's behavior, emotion, motivation, and thought pattern characteristics. The automatic detection of personality traits has many important practical applications.

In recommender systems, the products and services recommended to a person should be those that have been positively evaluated by other users with a similar personality type. The Big Five is the most widely accepted model of personality.

François Mairesse and his colleagues used the Linguistic Inquiry and Word Count (LIWC) dataset and other features, such as imageability, to conduct automated personality detection on an essay dataset.[13]

Recently, Navonil Majumder and his colleagues presented a deep learning method to determine personality type from stream-of-consciousness essays by detecting the presence or absence of the Big Five traits in the author's psychological profile.[14] For each of the five traits, they trained a separate binary classifier, with identical architecture, based on a novel document representation technique. The classifier is implemented as a specially designed deep convolutional neural network (CNN), with injection of the document-level Mairesse features, extracted directly from the text, into an inner layer.

In recommender systems, the products and services recommended to a person should be those that have been positively evaluated by other users with a similar personality type.

The first layers of the network treat each sentence of the text separately, then aggregate sentences into a document vector.

Sarcasm Detection

Sarcasm is always directed at someone or something. A target of sarcasm is the person or object against whom or which the utterance is directed. Targets can be the sender, the addressee, or a third party (or a combination of the three). The presence of sarcastic sentences may completely change the meaning of a review, therefore misleading the interpretation of its polarity.

Although the use of irony and sarcasm is well studied from its linguistic and psychologic aspects, sarcasm detection is still represented by very few works in the computational literature.

To date, most approaches to sarcasm detection have treated the task primarily as a text categorization problem. Sarcasm, however, can be expressed in subtle ways and requires a deeper understanding of natural language that standard text categorization techniques cannot grasp. Livia Polanyi and Annie Zaenen suggested a theoretical framework in which the context of sentiment words shifts the valence of the expressed sentiment.[15] This assumes that, although most salient clues about attitude are provided by the writer's lexical choice, the text's organization also provides relevant information for assessing attitude. More recently, researchers developed deep models based on a pretrained CNN for extracting sentiment, emotion and personality features for sarcasm detection.[16]

Metaphor Understanding

Metaphors are commonly used to substitute complex concepts with simple concepts that bear similar ideas but are not literally applicable.

In the context of sentiment analysis, metaphor detection and understanding are necessary for aspect extraction and polarity detection. The aspects of an undetected metaphor, in fact, could be classified as off-topic and, hence, reduce the accuracy of sentiment classification.

Early works employed handcrafted rules and knowledge bases for metaphor detection, but they suffered from scalability issues.

More recent approaches tried to leverage either conceptual metaphor mappings or selectional preferences. Both require extensive knowledge of the mappings/preferences in question, as well as sufficient data for all involved conceptual domains. Creating these resources is expensive and often limits the scope of these systems. Recently, Marc Schulder and Eduard Hovy proposed a statistical approach to metaphor detection that requires no knowledge of semantic concepts or the metaphor's source domain.[17] The model utilizes the rarity of novel metaphors, marking words that do not match a text's typical vocabulary as metaphor candidates.

Aspect Extraction

In opinion mining, different levels of analysis granularity have been proposed, each having advantages and drawbacks. Aspect-based sentiment analysis focuses on the relations between aspects and document polarity.

Aspects are opinion targets, that is, the specific features of a product or service that users like or dislike. For example, the sentence, "The screen of my phone is really nice and its resolution is superb," expresses a positive polarity about the phone under review. More specifically, the opinion holder is expressing a positive polarity about its *screen* and *resolution*; these concepts are thus called opinion targets, or aspects. It is important to identify aspects because reviewers might express opposite polarities about different aspects in the same sentence. Early approaches to aspect extraction included linguistic rules based on statistical observations, pointwise mutual information between noun phrase and product class, topic modeling, and more. Recently, researchers developed a hybrid approach to aspect extraction, which involved the use of a seven-layer-deep CNN, for tagging each word in opinionated sentences as either an aspect or nonaspect word, in concomitance with aspect-specific linguistic patterns.[18]

Polarity Detection

Polarity detection is the most popular sentiment analysis task. In fact, many research works even use the terms "polarity detection" and "sentiment analysis" interchangeably. This is due to the (limited) definition of sentiment analysis as the NLP task that aims to categorize a piece of text as either positive or negative.

In fact, early approaches simply focused on this binary classification (often ignoring the presence of neutral content) by employing knowledge bases, rule-based classification, and supervised learning. Later works focused on a finer-grained categorization that included measuring the intensity of the polarity detected.[19]

Recent approaches to polarity detection include deep learning techniques (for example, CNNs and LSTMs relying on constituency parsing trees) and hybrid frameworks that leverage an ensemble of linguistics, knowledge representation, and machine learning to achieve a better understanding of the contextual role of each concept within the sentence, by allowing sentiments to flow from concept to concept based on dependency relations.[20]

Recent developments in machine learning have enabled the field of NLP to make great progress. Some NLP tasks, however, require more than a mere data-driven approach to achieve human-like performance. Sentiment analysis is one of them as it entails several NLP problems, including word sense disambiguation, anaphora resolution, sarcasm detection, metaphor understanding, and aspect extraction. In this article, we offered an overview of such NLP problems and provided some guidelines on how and why these should be concatenated. We hope this will serve as an eye-opener to those who believe that sentiment analysis is simply a binary classification task and, hence, pave the path to an ensemble approach to NLP that leverage both data-driven (bottom-up) algorithms and theory-driven (top-down) methods to mimic the way humans decode and understand natural language. ■

References

1. E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, 2014, pp. 48–57.
2. T. Bertaglia and M. das Graças, "Exploring Word Embeddings for Unsupervised Textual User-Generated Content Normalization," *Proc. 26th Int'l Conf. Computational Linguistics* (COLING 16), 2016, pp. 112–120.
3. J.C. Reynar and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," *Proc. 5th Conf. Applied Natural Language Processing*, 1997, pp. 16–19.
4. Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv preprint arXiv:1508.01991, 2015.
5. L.A Ramshaw and M.P. Marcus, "Text Chunking Using Transformation-Based

Learning," *Natural Language Processing Using Very Large Corpora*, Springer, 1999, pp. 157–176.
6. M. Kestemont et al., "Lemmatization for Variation-Rich Languages Using Deep Learning," *Literary and Linguistic Computing*, 2016, p. fqw034.
7. Y.K. Lee and H.T. Ng, "An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation," *Proc. ACL-02 Conf. Empirical Methods in Natural Language Processing*, vol. 10, 2002, pp. 41–48.
8. E. Cambria et al., "SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives," *Proc. 26th Int'l Conf. Computational Linguistics* (COLING 16), 2016, pp. 2666–2677.
9. Y. Ma, E. Cambria, and S. Gao, "Label Embedding for Zero-Shot Fine-Grained Named Entity Typing," *Proc. 26th Int'l Conf. Computational Linguistics* (COLING 16), 2016, pp. 171–180.
10. X. Zhong, A. Sun, and E. Cambria, "Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules," *Proc. 55th Ann. Meeting Assoc. for Computational Linguistics* (ACL 17), 2017, pp. 420–429.
11. R. Mitkov, R. Evans, and C. Orasan, "A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method," *Computational Linguistics and Intelligent Text Processing*, Springer, 2002, pp. 168–186.
12. I. Chaturvedi et al., "Bayesian Network Based Extreme Learning Machine for Subjectivity Detection," *J. Franklin Inst.*, 2017; doi:10.1016/j.jfranklin.2017.06.007.
13. F. Mairesse et al., "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text," *J. Artificial Intelligence Research* (JAIR), vol. 30, Sept.–Dec. 2007, pp. 457–500.
14. N. Majumder et al., "Deep Learning-Based Document Modeling for Personality Detection from Text," *IEEE Intelligent Systems*, vol. 32, no. 2, 2017, pp. 74–79.
15. L. Polanyi and A. Zaenen, "Contextual Valence Shifters," *Computing Attitude and Affect in Text: Theory and Applications*, Springer, 2006, pp. 1–6.
16. S. Poria et al., "A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks," *Proc. 26th Int'l Conf. Computational Linguistics* (COLING 16), 2016, pp. 1601–1612.
17. M. Schulder and E. Hovy, "Metaphor Detection Through Term Relevance," *Proc. 2nd Workshop on Metaphor in NLP*, 2014, pp. 18–26.
18. S. Poria et al., "Convolutional MKL Based Multimodal Emotion Recognition and Sentiment Analysis," *Proc. IEEE Int'l Conf. Data Mining* (ICDM 16), 2016, pp. 439-448.
19. M. Thelwall, "The Heart and Soul of the Web? Sentiment Strength Detection in the Social Web with SentiStrength," *Cyberemotions: Collective Emotions in Cyberspace*, J.A. Holyst, ed., Springer, 2017, pp. 119–134.
20. S. Poria et al., "Sentiment Data Flow Analysis by Means of Dynamic Linguistic Patterns," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, 2015, pp. 26–36.

Erik Cambria is an assistant professor in the School of Computer Science and Engineering at Nanyang Technological University. He is associate editor of several journals and is involved in several international conferences as program committee member, workshop organizer, and program chair. Contact him at cambria@ntu.edu.sg.

Soujanya Poria is a research scientist at Temasek Laboratories, Nanyang Technological University. His research interests include computational linguistics, deep learning, and multimodal sentiment analysis. Poria holds a PhD in computer science and mathematics from the University of Stirling. Contact him at sporia@ntu.edu.sg.

Alexander Gelbukh is a research professor at the Centro de Investigación en Computación of the Instituto Politécnico Nacional. His research interests include computational linguistics, natural language processing, and sentic computing. He is a member of the Mexican Academy of Sciences and founding member of the Mexican Academy of Computing. Contact him at gelbukh@cic.ipn.mx.

Mike Thelwall is a professor of information science and leads the Statistical Cybermetrics Research Group at the University of Wolverhampton, UK. He has developed and evaluated free software and methods for systematically gathering and analyzing social web data, including for sentiment analysis, altmetrics, and webometrics. Contact him at m.thelwall@wlv.ac.uk.