Phonetic-Based Microtext Normalization for Twitter Sentiment Analysis

Ranjan Satapathy, Claudia Guerreiro, Iti Chaturvedi, Erik Cambria
Nanyang Technological University
50 Nanyang Ave, 639798, Singapore
Email: {satapathy.ranjan,claudiaguerreiro,iti,cambria}@ntu.edu.sg

Abstract—**The proliferation of Web 2.0 technologies and the increasing use of computer-mediated communication resulted in a new form of written text, termed microtext. This poses new challenges to natural language processing tools which are usually designed for well-written text. This paper proposes a phonetic-based framework for normalizing microtext to plain English and, hence, improve the classification accuracy of sentiment analysis. Results demonstrated that there is a high (>0.8) similarity index between tweets normalized by our model and tweets normalized by human annotators in 85.31% of cases, and that there is an accuracy increase of >4% in terms of polarity detection after normalization.**

Index Terms—**Text normalization, Error correction, Microtext analysis, Sentiment analysis, Twitter.**

1. Introduction

In recent years, the rise and expansion of social media enabled users to share their views, lives and interests in an impromptu manner and in real time. Since Web 2.0 is meant for human consumption, web users tend to abbreviate English terms by relying on the phonetic of numbers and letters. For example, they write terms or sentences such as "c u 2morrow" (see you tomorrow), "tgif" (thank God it's Friday) and "abt" (about) which may not be found in Standard English but are widely seen in short message service (SMS) texts, Twitter messages, Facebook updates, blogs, discussion forums and chat logs. In this way, computer-mediated communication has generated a slang often referred to as "microtext" which differs from well-written text [1].

Microtext became one of the most widespread communication forms among users due to its casual writing style and colloquial tone [2], plus its exponential growth is highly perceptible. For instance, according to CTIA [3], American people sent 196.9 billion text messages in 2011 compared to 12.5 billion in 2006. Another statistic showed that until May 2016 there were nearly 500 million Tweets sent each day, meaning 6,000 Tweets every second [4]. Given that most data today is mined from the Web, microtext analysis is key for many natural language processing (NLP) and data mining tasks. In the context of social data analytics [5], especially, microtext normalization is a necessary step for pre-processing text before polarity detection is performed.

Some of the microtext key features are relaxed spelling and reliance on emoticons and out-of-vocabulary (OOV) words involving phonetic spelling (e.g., 'b4' for 'before'), emotional emphasis (e.g., 'goooooood' for 'good') and popular acronyms (e.g., 'otw' for 'on the way') [6], [1], [7]. The challenge arises when trying to automatically rectify and replace them with the correct in-vocabulary (IV) words [8], [9]. It could be thought that microtext normalization is as a simple as performing find-and-replace pre-processing [10]. However, the wide-ranging diversity of spellings makes this solution impractical (e.g., the spelling of the word "tomorrow" includes tom, 2moro, 2m, 2ma, 2maro, 2mmrw, 2mo, 2mora, 2moro, 2morow, 2morro, 2morrow, 2moz, 2mozz, 2mro, 2mrw, 2mw and 2mz, among others). Furthermore, given the productivity of users, novel forms which are not bound to orthographic norms in spelling can emerge. For instance, a sampling of Twitter studied in [8] found over 4 million OOV words where new spellings were created constantly, both voluntarily and accidentally.

This paper proposes a novel framework to deal with microtext normalization in Twitter in a human-inspired way, i.e., by shifting to the phonetic domain to better decode microtext. Humans, in fact, are able to understand abbreviations and uncanny spellings they have never seen before because they automatically shift to the phonetic domain when they read text. To this end, we propose an ensemble approach that leverages a phonetic algorithm for normalizing OOV words and tries to handle the rest (e.g., emoticons and acronyms) using a lexicon.

The rest of the paper is organized as follows: Section 2 presents related work in the field of sentiment analysis and text normalization; Section 3 describes the proposed model; Section 4 proposes experimental results; finally, Section 5 provides concluding remarks.

2. Related Work

Works related to the proposed model fall under two main categories: sentiment analysis and microtext analysis.

2.1. Sentiment Analysis

Since the beginning of human history, people have been considered by nature as social animals who are highly sus-

2375-9259/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDMW.2017.59

ceptible to opinions as practically all undertakings and behaviors are influenced by them. Accordingly, when choices are to be taken, individuals and organizations frequently look for others' opinions. Opinions and its associated concepts such as sentiments, emotions, attitudes, and evaluations are the focuses of study of sentiment analysis.

Sentiment analysis [11] is a branch of affective computing research [12] that aims to classify text (but sometimes also audio and video [13]) into either positive or negative (but sometimes also neutral [14]). While most works approach it as a simple categorization problem, sentiment analysis is actually a suitcase research problem [15] that requires tackling many NLP sub-tasks, including aspect extraction [16], named entity recognition [17], word polarity disambiguation [18], temporal tagging [19], personality recognition [20], and sarcasm detection [21].

[22], [23] state that sentiment analysis has numerous applications with many purposes such as the detection of the mood of the market based on specialists' opinions [24], [25], the analysis of customers' reviews about products or services [26], [27], the analysis of touristic sites through tourists' comments [28], and the analysis of politicians [29] or topics connected to politics [30].

By itself, sentiment analysis systems are often characterized into statistics-based and knowledge-based systems [31]. On the one hand, statistical approaches have proven to be generally semantically feeble [32]. This is due to the fact that statistical text classifiers only work with adequate precision when given a satisfactorily vast text input [33]. Although statistical approaches are able to affectively classify users' text on the page or section level, they do not work properly on smaller text parts such as sentences.

On the other hand, concept-level sentiment analysis is a task which relies on large semantic knowledge bases which has recently growing interest within the scientific community as well as the business world. It emphasizes on a semantic text analysis through the use of web ontologies or semantic networks which enable an aggregation of the conceptual and affective information associated with natural language opinions [34], [35], [36], [37].

The analysis at concept level enables to infer the semantic and affective information associated with natural language opinions and, well ahead, to enable a comparative fine-grained feature-based sentiment analysis. Henceforth, the approach proposed relies on concept-level sentiment analysis because it leaves behind the sightless use of keywords and word co-occurrence counts rather depending on the implied features linked with natural language concepts.

2.2. Microtext Analysis

Microtext has become ubiquitous in today's communication. 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 [38].

In [39], authors present a novel unsupervised method to translate Chinese abbreviations. It automatically extracts the relation between a full-form phrase and its abbreviation from monolingual corpora, and induces translation entries for the abbreviation by using its full-form as a bridge. [40] uses a classifier to detect out of vocabulary words, and generates correction candidates based on morphophonemic similarity. While existing studies have developed different microtext normalization techniques, academics have not reached a consensus to resolve this issue. Normalization has mostly been handled through three well-known NLP tasks, namely: spelling correction, statistical machine translation (SMT) and automatic speech recognition (ASR).

2.2.1. Spelling Correction. Correction is executed on a word-per-word basis seen as a spelling checking task. This model gained extensive attention in the past and a diversity of correction practices have been endorsed by [41], [42], [43], [44], [45]. Instead, [46] and [47] proposed a categorization of abbreviation, stylistic variation, prefix-clipping, which was then used to estimate their probability of occurrence. Thus far, the spelling corrector became widely popular in the context of SMS messages, where [48] advanced the hidden Markov model whose topology takes into account both "graphemic" variants (e.g., typos, omissions of repeated letters, etc.) and "phonemic" variants (e.g., spellings that resemble the word's pronunciation). However, all the above work only focused on the normalization of words without considering their respective context.

2.2.2. Statistical Machine Translation. SMT outlooks microtext as a foreigner language that has to be translated to plain English, meaning that normalization is done through a SMT task. When compared to the previous task, this method appears to be rather straightforward and better since it has the possibility to model (context-dependent) one-to-many relationships which were out-of-reach previously [49]. Some examples of works include [50], [51], [52]. However, the SMT still overlooks some features of the task, particularly the fact that lexical creativity verified in social media messages is barely captured in a stationary sentence board.

2.2.3. Automatic Speech Recognition. ASR considers that microtext tends to be a closer approximation of the word's phonemic representation rather than its standard spelling. As follows, the key of microtext normalization becomes very similar to speech recognition which consists of decoding a word sequence in a (weighted) phonetic framework. For example, [49] proposed to handle normalization based on the observation that text messages present a lot of phonetic spellings, while more recently [10] proposed an algorithm to determine the probable pronunciation of English words based on their spelling. Although the computation of a phonemic representation of the message is extremely valuable, it does not solve entirely all the microtext normalization challenges (e.g., acronyms and misspellings do not resemble their respective IV words' phonemic representation).

Perhaps the best normalization approach might be a combination of these methods, similarly to what [53] has done by merging the advantages of SMT and the spelling corrector model. Similarly, this paper proposes an ensemble approach that normalizes both phonetics and fixed expressions (e.g., acronyms and emotions) in the same way as a human reader would do.

3. Proposed Model

The proposed model handles acronyms and emoticons using a lexicon-based approach first (Section 3.1) and later attempts to process what is left un-normalized using a phonetic algorithm (Section 3.2). The final output of such modules is fed to a polarity classifier (Section 3.3).

3.1. Lexicon-Based Approach

We extracted acronyms and emoticons from various online sources. In particular, we crawled popular acronyms from NetLingo[1], MakeUseOf[2], Acronyms and Slang[3], and Internet Slang[4]. Common emoticons, instead, were crawled from Cool Smileys[5], Internet Slang[6], and Fbicons[7]. After removing duplicates and IV words, we ended up with a lexicon containing 1,727 acronyms and 512 emoticons. Table 1 shows a sample of the lexicon which has been used in this paper.

TABLE 1. SAMPLE OF THE LEXICON

OOV form	IV form	Polarity
ygbk	you gotta be kidding	negative
ykwim	you know what i mean	neutral
ykw	you know what	neutral
ulkgr8	you look great	positive
ymak	you may already know	neutral
ymal	you might also like	positive
ymbkm	you must be kidding me	negative
ynk	you never know	neutral
uok	you ok	neutral
:)	smiling	positive
:-)	smiling	positive
:-]	smiling	positive
:D	laughing	positive
:-D	laughing	positive
8D	laughing	positive
:‘(	crying	negative
:‘-(	crying	negative
:(	frowning	negative
:-(	frowning	negative
:c	frowning	negative

3.2. Phonetic-Based Approach

We used a simple but effective algorithm to handle microtext based on its phonetics: Soundex [54]. Soundex is largely used to group similar sounding letters together and assign each group a numerical number. The main goal of Soundex is to use homophones for encoding text with a numerical representation, which can be easily matched with other similar sounding characters having the same numerical code. This results in retrieving a list of words that are pronounced similarly with very little variation in their homophones [55], [56]. We chose Soundex in spite of other phonetic algorithms due to its following advantageous features:

1) It is simple and intuitive to operate;
2) The processing time is fairly short;
3) It has a high tolerance for discrepancies in words that sound very similar or are identical.

Soundex hash value is calculated by using the first letter of a name and converting its consonants to digits through a simple lookup table. The pseudo-code of the algorithm is as follows:

```
1) Retain the first letter of the word;
2) Change all occurrences of the following letters to
   ‘0’ (zero): ‘A’, ‘E’, ‘I’, ‘O’, ‘U’, ‘H’, ‘W’, ‘Y’;
3) Change letters to digits as follows:
    a) B, F, P, V → 1
    b) C, G, J, K, Q, S, X, Z → 2
    c) D,T → 3
    d) L → 4
    e) M, N → 5
    f) R → 6
4) Remove all pairs of consecutive digits;
5) Remove all zeros from the resulting string;
6) Pad the resulting string with trailing zeros and
   return the first four positions, which will be of the
   form ⟨uppercase letter⟩ ⟨digit⟩ ⟨digit⟩ ⟨digit⟩.
```

In this work, we used Soundex to normalize OOV words by phonetic code matching on a specific knowledge base. Since our final goal is to perform sentiment analysis, we chose SenticNet [33], a sentiment lexicon that contains polarity scores of both single words and multi-word expressions. In particular, all SenticNet concepts were assigned a specific Soundex code so that later any OOV word found in text could be normalized according to such a code (if present in SenticNet).

For example, the OOV word ‘cooooooooool’ would be firstly converted by Soundex to C400, and secondly such a code would be used to find a match in SenticNet, i.e. ‘cool’ (which also has Soundex code C400). Soundex, however, is unable to decode the phonetics of numbers in between letters, e.g., ‘2n8’ for ‘tonight’. To this end, we replace numbers in between letters as their literal equivalent, e.g., ‘2n8’→‘twoneight’→T523→‘tonight’. Before we do that, however, we check for the presence of emoticons as some of them contain numbers, e.g., ‘3’ in ‘<3’.

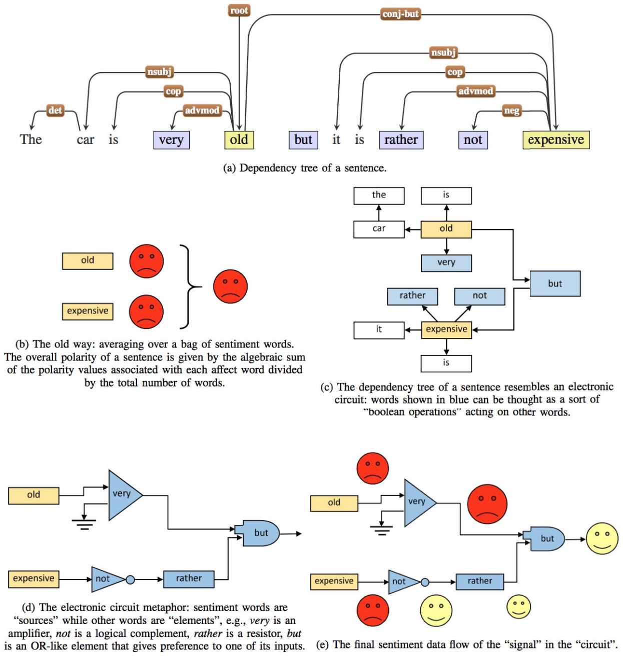
[1] http://netlingo.com
[2] http://makeuseof.com/tag/30-trendy-internet-acronyms
[3] http://acronymsandslang.com
[4] http://internetslang.com
[5] http://cool-smileys.com/text-emoticons
[6] http://internetslang.com/list.asp?i=other
[7] http://fbicons.net

3.3. Polarity Detection

Existing approaches to sentiment analysis mainly rely on parts of text in which opinions are explicitly expressed like polarity terms, affect words and their co-occurrence frequencies. However, opinions and sentiments are often conveyed implicitly through latent semantics, making purely syntactic approaches ineffective. To this end, we used sentic computing [57], a novel approach that is able to capture latent information in terms of semantics and sentics, i.e., the denotative and connotative information commonly associated with real-world objects, actions, events, and people.

Verb and noun concepts in SenticNet are linked to primitives so that, for example, concepts such as *eat_rice* or *slurp_noodles* are generalized as *INGEST_FOOD*. In this way, most concept inflections can be captured by the knowledge base: verb concepts like *eat*, *slurp*, *munch* are all represented by their conceptual primitive *INGEST* while noun concepts like *steak*, *rice*, *noodles* are replaced with their ontological parent *FOOD*.

SenticNet steps away from blindly using keywords and word co-occurrence counts, relying on the implicit meaning associated with commonsense concepts instead. Superior to purely syntactic techniques, SenticNet can detect subtly expressed sentiments by enabling the analysis of multiword expressions that do not explicitly convey emotion, but are instead related to concepts that do so. Yet, the study of emotions is one of the most confused (and still open) chapters in the history of psychology. This is mainly due to the ambiguity of natural language, which does not facilitate the description of mixed emotions in an unequivocal way.



(a) Dependency tree of a sentence.

(b) The old way: averaging over a bag of sentiment words. The overall polarity of a sentence is given by the algebraic sum of the polarity values associated with each affect word divided by the total number of words.

(c) The dependency tree of a sentence resembles an electronic circuit: words shown in blue can be thought as a sort of "boolean operations" acting on other words.

(d) The electronic circuit metaphor: sentiment words are "sources" while other words are "elements", e.g., *very* is an amplifier, *not* is a logical complement, *rather* is a resistor, *but* is an OR-like element that gives preference to one of its inputs.

(e) The final sentiment data flow of the "signal" in the "circuit".

Figure 1. An example of sentic patterns

The Hourglass of Emotions [58] applied in SenticNet organizes primary emotions around four independent but concomitant dimensions, whose different levels of activation make up the total emotional state of the mind. Such a reinterpretation is inspired by Minsky's theory of the mind, according to which brain activity consists of different independent resources and that emotional states result from turning some set of these resources on and another off [59]. This way, the model can potentially synthesize the full range of emotional experiences in terms of Pleasantness, Attention, Sensitivity, and Aptitude, as the different combined values of the four affective dimensions can also model affective states we do not have a specific name for, due to the ambiguity of natural language and the elusive nature of emotions.

The right way to use SenticNet for the task of polarity detection is in conjunction with sentic patterns [60]. Sentic patterns are sentiment-specific linguistic patterns that infer polarity by allowing affective information to flow from concept to concept based on the dependency relation between clauses. The main idea behind such patterns can be best illustrated by analogy with an electronic circuit, in which few 'elements' are 'sources' of the charge or signal, while many elements operate on the signal by transforming it or combining different signals (Fig. 1). This implements a rudimentary type of semantic processing, where the 'meaning' of a sentence is reduced to only one value: its polarity.

4. Experiments

This section proposes a model's evaluation in terms of both similarity of microtext normalization with respect to a human annotator (Section 4.1) and improved accuracy of polarity detection (Section 4.2). A dataset of approximately 4,000 tweets (randomly selected) was constructed and annotated by three reviewers (Cohen's kappa = 0.78) in terms of polarity (positive or negative). We also asked one reviewer to normalize each of the 4,000 tweets to plain English (Table 2).

TABLE 2. MANUAL NORMALIZATION OF TWEETS

Original tweet	Manually-normalized tweet
nt vegan nemor lol	not vegan anymore laughing out loud
how are you so gorgeousss omg omg	how are you so gorgeous oh my god oh my god
pls start sayin dis prayer	please start saying this prayer
idk why its soooo goood	I don't know why it's so good

4.1. Similarity Evaluation

The Ratcliff/Obershelp pattern-matching algorithm [61] has been used as an evaluation technique for the microtext normalization. Ratcliff/Obershelp algorithm is able to return a percentage to show how alike two strings are. The main advantage of this algorithm is that it enables the recognition of matches in substrings quickly and easily.

In particular, we used Soundex to codify the phonetics of both tweets normalized by our model and tweets normalized by humans. Then, the pattern-matching algorithm has been used to calculate the similarity of the resulting Soundex codes (Fig. 2). 85.31% of texts have a similarity index equal to or greater than 0.8. Table 3 shows some examples.

TABLE 3. RESULTS OF SEQUENCE MATCHER ALGORITHM, COMPARING SOUNDEX OUTPUT

Proposed model	**Annotated text**	**Similarity**
R3T63I2M5N3W5-W0Y4F23T2I35325	R3T63I2M5N3W5-W0Y4F23T2I35325	1.0
D53R16M0T0B4H0-I32T0E21521	D53R16M0T0B4H0-I32T0E21521	1.0
I0W2L25F6S5352T-0F163B3T6W2N35-2L25234I0W2T65-23L1A0H53	I0W2L252F6S5352T-0F163B3T6W2N35-2L25234I0W2T65-23L1A0H53	0.98
R3W4A42G5B0-T0H5	R3W4A42G523B0-T0H5	0.89
O523Y52M624G3T-62T0T0F542O1B23	O523Y52M624G3T-62T0T0F542O1-B63523	0.88

4.2. Polarity Evaluation

The main goal of our model was to apply microtext normalization to improve the performance of sentiment analysis. As most polarity classifiers are optimized for plain English, in fact, we expected polarity classification to improve after tweets are normalized from informal to formal text. To this end, we processed both original tweet and normalized tweet with Sentic API[8] and compared results against the human-annotated polarity labels (Fig. 3). Table 4 shows a sample output of the polarity detection module for each tweet.

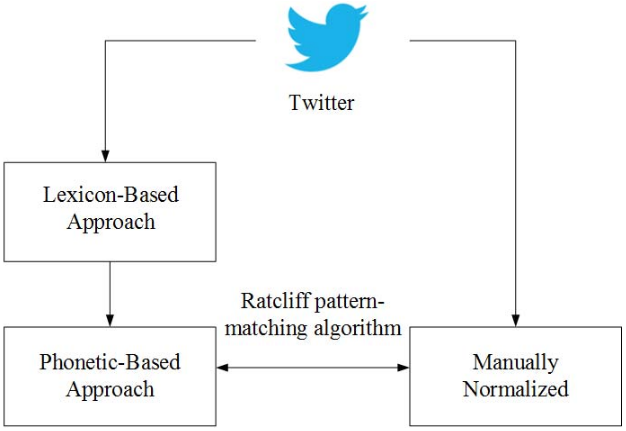


Figure 2. Similarity evaluation process

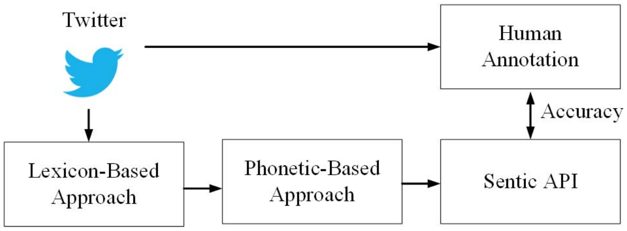


Figure 3. Polarity evaluation process

It can be observed from the table how, in some cases, the polarity of each tweet changes before and after microtext analysis. The application of microtext normalization module results in an accuracy increase of >4% (from 77.47% to 81.59%) in terms of polarity detection.

TABLE 4. RESULTS OF POLARITY DETECTION MODULE

Original Tweet	**Polarity before microtext analysis**	**Polarity after microtext analysis**
dance parties with hayley and levi are probs my fav thing ever !	positive	positive
today was fun someone could of made it better lol me gelo and alex was really clowning	positive	positive
so ima keep on drinking cause i luv this shit	negative	positive
dads gonna buy stronger chillis so i can do it against one of my siblings lewl	negative	positive
tent erected in the middle of a main road in phase 4 in mamelodi , we knw they r grieving bt not in main road .	positive	negative
you should check out in wan chai . considered one of da most awsome burgers in hk	negative	positive
hey m8 . whr we goin 2n8	negative	positive

5. Conclusion

Social media language is considerably different from other written text. Many of the efforts to illustrate and overcome this discrepancy have focused on normalization. In this paper, a novel framework was proposed to deal with microtext normalization in Twitter for sentiment analysis purposes. The proposed model combined two different methods for microtext normalization (namely, lexicon- and phonetic-based) and Sentic API as a polarity classifier.

[8] http://sentic.net/api

Ratcliff/Obershelp pattern-matching algorithm was seen as an apt evaluation technique. The results demonstrated that 85.31% of texts have a similarity index equal to or greater than 0.8, showing that this framework has the ability to correctly handle many types of normalization challenges. Moreover, the proposed model had the ability to enhance the accuracy of polarity detection from 77.47% to 81.59%.

Future work will focus on expertimenting whether Soundex could be replaced with a more complex phonetic system (e.g., IPA) in order to improve the generalization of the proposed rules. We also plan to employ deep learning to learn new forms of microtext by lexical substitution.

References

[1] K. D. Rosa and J. Ellen, "Text classification methodologies applied to micro-text in military chat," in *Proc. Eight International Conference on Machine Learning and Applications*, Miami, 2009, pp. 710–714.

[2] F. Liu, F. Weng, and X. Jiang, "A Broad-Coverage Normalization System for Social Media Language," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, no. July, 2012, pp. 1035–1044.

[3] C. Li and Y. Liu, "Normalization of Text Messages Using Character- and Phone-based Machine Translation Approaches," *INTERSPEECH*, pp. 2330–2333, 2012.

[4] Brandwatch, "44 Twitter Statistics for 2016," 2016. [Online]. Available: https://www.brandwatch.com/blog/44-twitter-stats-2016/

[5] E. Cambria, H. Wang, and B. White, "Guest editorial: Big social data analysis," *Knowledge-Based Systems*, vol. 69, pp. 1–2, 2014.

[6] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL student research workshop*. Association for Computational Linguistics, 2005, pp. 43–48.

[7] Z. Xue, D. Yin, and B. D. Davison, "Normalizing Microtext," *Analyzing Microtext*, pp. 74–79, 2011.

[8] F. Liu, F. Weng, B. Wang, and Y. Liu, "Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision," *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 71–76, 2011.

[9] S. Petrović, M. Osborne, and V. Lavrenko, "The Edinburgh Twitter corpus," in *Proceedings of the NAACL HLT Workshop on Computational Linguistics in a World of Social Media*, 2010, pp. 25–26.

[10] R. Khoury, "Microtext Normalization using Probably-Phonetically-Similar Word Discovery," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on.*, 2015, pp. 392–399.

[11] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*. Cham, Switzerland: Springer, 2017.

[12] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Information Fusion*, vol. 37, pp. 98–125, 2017.

[13] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *ICDM*, Barcelona, 2016, pp. 439–448.

[14] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, and E. Cambria, "Bayesian network based extreme learning machine for subjectivity detection," *Journal of The Franklin Institute*, 2017.

[15] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE Intelligent Systems*, vol. 32, no. 6, 2017.

[16] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.

[17] Y. Ma, E. Cambria, and S. Gao, "Label embedding for zero-shot fine-grained named entity typing," in *COLING*, Osaka, 2016, pp. 171–180.

[18] Y. Xia, E. Cambria, A. Hussain, and H. Zhao, "Word polarity disambiguation using bayesian model and opinion-level features," *Cognitive Computation*, vol. 7, no. 3, pp. 369–380, 2015.

[19] X. Zhong, A. Sun, and E. Cambria, "Time expression analysis and recognition using syntactic token types and general heuristic rules," in *ACL*, 2017, pp. 420–429.

[20] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 74–79, 2017.

[21] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," in *COLING*, 2016, pp. 1601–1612.

[22] A. L. F. Alves, C. d. S. Baptista, A. A. Firmino, M. G. de Oliveira, and A. C. de Paiva, "A Spatial and Temporal Sentiment Analysis Approach Applied to Twitter Microtexts," *Journal of Information and Data Management*, vol. 6, no. 2, p. 118, 2016.

[23] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[24] M. Koppel and I. Shtrimberg, "Good news or bad news? Let the market decide," in *Computing attitude and affect in text: Theory and applications*. Springer Netherlands, 2006, pp. 297–301.

[25] N. O'Hare, M. Davy, A. Bermingham, P. Ferguson, P. á. Sheridan, C. Gurrin, and A. F. Smeaton, "Topic-dependent sentiment analysis of financial blogs," *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, no. November, pp. 9–16, 2009.

[26] M. Eirinaki, S. Pisal, and J. Singh, "Feature-based opinion mining and ranking," *Journal of Computer and System Sciences*, vol. 78, no. 4, pp. 1175–1184, 2012.

[27] M. Hu and B. Liu, "Mining and summarizing customer reviews," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, 2004.

[28] E. Bjørkelund, T. H. Burnett, and K. Nørvåg, "A study of opinion mining and visualization of hotel reviews," *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, pp. 229–238, 2012.

[29] R. Awadallah, M. Ramanath, and G. Weikum, "PolariCQ: Polarity classification of political quotations," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 1945–1949.

[30] Y. Fang, L. Si, N. Somasundaram, and Z. Yu, "Mining contrastive opinions on political texts using cross-perspective topic model," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 63–72.

[31] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[32] E. Cambria and B. White, "Jumping NLP curves: a review of natural language processing research," *IEEE Computational Intelligence*, vol. 9, no. 2, pp. 48–57, 2014.

[33] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives," in *COLING*, 2016, pp. 2666–2677.

[34] M. Araújo, P. Gonçalves, and M. Cha, "iFeel: a system that compares and combines sentiment analysis methods," in *WWW*, 2014, pp. 75–78.

[35] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Meta-level sentiment models for big social data analysis," *Knowledge-Based Systems*, vol. 69, pp. 86–99, 2014.

[36] G. Gezici, R. Dehkharghani, B. Yanikoglu, D. Tapucu, and Y. Saygin, "Su-sentilab: a classification system for sentiment analysis in Twitter," in *Proceedings of the International Workshop on Semantic Evaluation, Atlanta*, 2013, pp. 471–477.

[37] D. R. Recupero, V. Presutti, S. Consoli, and A. N. Gangemi, "Sentilo: frame-based sentiment analysis," *Cognitive*, vol. 7, no. 2, pp. 211–225, 2015.

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

[39] Z. Li and D. Yarowsky, "Unsupervised translation induction for chinese abbreviations using monolingual corpora," in *In Proceedings of ACL/HLT*, 2008.

[40] B. Han and T. Baldwin, "Lexical normalisation of short text messages: Makn sens a# twitter," in *ACL*, 2011, pp. 368–378.

[41] K. W. Church and W. A. Gale, "Probability scoring for spelling correction," *Statistics and Computing*, vol. 1, no. 2, pp. 93–103, 1991.

[42] E. Brill and R. C. Moore, "An improved error model for noisy channel spelling correction," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000, pp. 286–293.

[43] M. Li, Y. Zhang, M. Zhu, and M. Zhou, "Exploring distributional similarity based models for query spelling correction," in *ACL*, 2006, pp. 1025–1032.

[44] D. L. Pennell and Y. Liu, "A Character-Level Machine Translation Approach for Normalization of SMS Abbreviations," in *IJCNLP*, 2011, pp. 974–982.

[45] K. Toutanova and R. C. Moore, "Pronunciation modeling for improved spelling correction," in *ACL*, 2002, pp. 144–151.

[46] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words," *Computer speech & language*, vol. 15, no. 3, pp. 287–333, 2001.

[47] P. Cook and S. Stevenson, "An unsupervised model for text message normalization," in *Proceedings of the workshop on computational approaches to linguistic creativity*, 2009, pp. 71–78.

[48] M. Choudhury, R. Saraf, V. Jain, S. Sarkar, and A. Basu, "Investigation and modeling of the structure of texting language," *International Journal of Document Analysis and Recognition*, vol. 10, no. 3-4, pp. 157–174, 2007.

[49] C. Kobus, F. Yvon, and G. é. Damnati, "Normalizing SMS: are two metaphors better than one?" in *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1. Association for Computational Linguistics, 2008, pp. 441–448.

[50] A. Aw, M. Zhang, J. Xiao, and J. Su, "A phrase-based statistical model for SMS text normalization," in *ACL*, 2006, pp. 33–40.

[51] M. Kaufmann and J. Kalita, "Syntactic normalization of Twitter messages," *natural language processing, Kharagpur, India*, 2010.

[52] D. L. Pennell and Y. Liu, "Normalization of informal text," *Computer Speech & Language*, vol. 28, no. 1, pp. 256–277, 2014.

[53] R. Beaufort, S. Roekhaut, L.-A. l. Cougnon, and C. d. Fairon, "A hybrid rule/model-based finite-state framework for normalizing SMS messages," in *ACL*. Association for Computational Linguistics, 2010, pp. 770–779.

[54] A. Beider and S. P. Morse, "Beider-Morse phonetic matching: An alternative to Soundex with fewer false hits," *Avotaynu: the International Review of Jewish*, 2008.

[55] Z. Bhatti, I. A. Ismaili, A. A. Shaikh, and W. Javaid, "Spelling error trends and patterns in sindhi," *CoRR*, vol. abs/1403.4759, 2014. [Online]. Available: http://arxiv.org/abs/1403.4759

[56] T. Naseem and S. Hussain, "Spelling error trends in urdu," in *Proceedings of Conference on Language Technology*, 2007.

[57] E. Cambria and A. Hussain, *Sentic computing: a common-sense-based framework for concept-level sentiment analysis*. Cham. Switzerland: Springer, 2015.

[58] E. Cambria, A. Livingstone, and A. Hussain, "The hourglass of emotions," *Cognitive behavioural systems*, pp. 144–157, 2012.

[59] M. Minsky, *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2007.

[60] S. Poria, E. Cambria, G. W. Winterstein, and G.-B. Huang, "Sentic patterns: Dependency-based rules for concept-level sentiment analysis." *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.

[61] J. Ratcliff and D. Metzener, "Pattern matching: The gestalt approach,," *Dr. Dobb's Journal*, 1988.