Sentiment Analyzer with Rich Features for Ironic and Sarcastic Tweets

Piyoros Tungthamthiti[1], **Enrico Santus**[2], **Hongzhi Xu**[2], **Chu-Ren Huang**[2], **and Kiyoaki Shirai**[1]

[1]Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi City, Ishikawa, Japan 923-1292
{s1320204,kshirai}@jaist.ac.jp

[2]Dept. of Chinese and Bilingual Studies
The Hong Kong Polytechnic University, Hong Kong
{e.santus,hongzhi.xu}@connect.polyu.hk
{churen.huang}@polyu.edu.hk

Abstract

Sentiment Analysis of tweets is a complex task, because these short messages employ unconventional language to increase the expressiveness. This task becomes even more difficult when people use figurative language (e.g. irony, sarcasm and metaphors) because it causes a mismatch between the literal meaning and the actual expressed sentiment. In this paper, we describe a sentiment analysis system designed for handling ironic and sarcastic tweets. Features grounded on several linguistic levels are proposed and used to classify the tweets in a 11-scale range, using a decision tree. The system is evaluated on the dataset released by the organizers of the SemEval 2015, task 11. The results show that our method largely outperforms the systems proposed by the participants of the task on ironic and sarcastic tweets.

1 Introduction

Whenever a message is encoded into linguistic form for being communicated – either in a spoken or written text[1] – information revealing judgments, evaluations, attitudes and emotions is also encoded (Martin and White, 2005). This is true for both informal and formal texts, independently of how much attention the writer pays in cleaning such information out. This is also true for texts posted on social networks (i.e. Facebook, Twitter, etc.), where judgments, evaluations, attitudes and emotions constitute an important part of the message (Pak and Paroubek, 2010).

Sentiment analysis (also known as opinion mining and subjectivity analysis) is a Natural Language Processing (NLP) task that focuses on identification of such judgments, evaluations, attitudes and emotions. It can be compared to other classification tasks, as it consists in associating the analyzed texts with a label that represents the sentiment of the message or the affective state of the writer (Hart, 2013).

In its earliest incarnations, sentiment analysis was limited to the identification of the polarity of the texts, and the classification label was either positive or negative. Later on, the task was extended to address more challenging and complex goals, such as the identification of the sentiment of the messages or the writer's affective state in a more fine scale, with labels including anger, happiness or depression.

Such extension could not avoid considering one of the most pervasive tools used in communication, namely figurative language. In fact, this expressive tool is not only very frequent in various kinds of texts, but it also strongly affects the sentiment expressed in the text, often completely reversing its polarity (Xu et al., 2015; Ghosh et al., 2015).

Because figurative language is used in unpredictable ways in communication (i.e. either in crystallized forms or in creative ways) and it can involve several linguistic and extra-linguistic levels (i.e. from syntax to concepts and pragmatics), its identification and understanding is often difficult, even for human beings. If humans are able to rely on prosody (e.g. stress or intonation), kinesis (e.g. facial gestures), co-text (i.e. immediate textual environment) and context (i.e. wider environment), as well as cultural background, machines cannot access the same type of information. These difficulties pose a major challenge in sentiment analysis.

Currently, a large number of studies have been devoting to the problem. Most of them focus on microblogging, especially Twitter, because i) social networks are rich of spontaneous public messages written by several users in different styles; ii)

[1] In this paper we will mainly refer to written texts, but most of what is said is also applicable to spoken ones.

29th Pacific Asia Conference on Language, Information and Computation pages 178 - 187
Shanghai, China, October 30 - November 1, 2015

tweets are short (i.e. a tweet can contain maximum 140 characters) and containing a lot of unconventional textual elements (e.g. emoticons, abbreviations, slang, emphasized capitalization and punctuation, etc.), which pose another interesting challenge; iii) social networks provide a precise picture of peoples' sentiments about a topic or product in a specific moment. This third point, in particular, is relevant for companies, political parties and other public entities in order to adapt and improve their marketing strategies and decisions (Medhat et al., 2014; Pang and Lee, 2008).

In this paper, we introduce a sentiment analysis system created with a particular focus on the identification and proper elaboration of irony and sarcasm in tweets. The system is developed by combining and improving two previous algorithms (Tungthamthiti et al., 2014; Xu et al., 2015). In particular, we propose a new method for coherence identification across sentences, some additional features indicating the strong emotion of the Twitter user, and several features of punctuations & special symbols that contribute to the final sentiment score.

2 Related work

Figurative language has been studied since the ancient Greece and Rome. It was, in fact, a part of the basic rhetorical background that every politician, lawyer and military officer should have had, in order to be able to persuade and convince his/her audience. Already in the first century CE, Quintilian (1953) defined irony as "saying the opposite of what you mean". This rhetorical figure violates the expectations of the listener, flouting the maxim of quality (Stringfellow, 1994; Grice, 1975). In a similar way, sarcasm is generally understood as the use of irony to mock or convey contempt (Stevenson, 2010). According to Haiman (1998), the main difference between sarcasm and irony is that sarcasm requires the presence of the intention to mocks. Irony, instead, can exist independently (e.g. there are ironic situations, but not sarcastic ones).

Although irony and sarcasm are well studied in linguistics and psychology, algorithms for their recognition and proper processing in sentiment analysis and other NLP tasks are still novel and far from perfect (Pang and Lee, 2008). In the last several years, however, such studies have attracted a lot of attention due to the availability of data. The simple use of hashtags on Twitter (e.g. #irony, #sarcasm or #not) allows the immediate collection of thousands of tweets. For example, 40,000 tweets were easily collected in four categories (i.e. irony, education, humour and politics) by Reyes et al. (2013).

Among the several approaches to irony and sarcasm in NLP, Carvalho et al. (2009) investigate the accuracy of a set of surface patterns (i.e. emoticons, onomatopoeic expressions for laughter, heavy punctuation marks, quotation marks and positive interjections) in comments at newspaper's articles. They show that surface patterns are much more accurate (from 45% to 85%) than deeper linguistic information.

Hao and Veale (2010) propose a nine steps algorithm to automatically distinguish ironic similes from non-ironic ones, without relying of any sentiment dictionary.

Tsur et al. (2010) propose a semi-supervised method for the automatic recognition of sarcasm in Amazon product reviews. Their method, which was compared to a strong heuristic baseline built by exploiting the star rating meta-data provided by Amazon (i.e. strongly positive reviews associated to low star rates were considered sarcastic), exploited syntactic and pattern-based features. A similar method, achieving high precision, was then applied to tweets (Davidov et al., 2010).

In Reyes and Rosso (2012), verbal irony is represented in terms of six kinds of features: n-grams, POS-grams, funny profiling, positive/negative profiling, affective profiling, and pleasantness profiling. They use Naive Bayesian, Support Vector Machine and Decision Tree classifiers, achieving an acceptable level of accuracy. Moreover, they built a freely available data set with ironic reviews from news articles, satiric articles and customer reviews, collected from Amazon.

More recently, a new complex model for identifying sarcasm was defined to extend the method far beyond the surface of the text and took into account features on four levels: signatures, degree of unexpectedness, style, and emotional scenarios (Reyes et al., 2013). They demonstrate that these features do not help the identification of irony and sarcasm in isolation. However, they do when they are combined in a complex framework.

Barbieri and Saggion (2014) use several lexical and semantic features, such as frequency of the words in reference corpora, their intensity, their written/spoken nature, their length and the number of related synsets in WordNet (Miller, 1995).

Buschmeier et al. (2014) provided an important baseline for irony detection in English by assessing the impact of features used in previous studies and evaluating them with several classifiers. They reach an F1-measure of up to 74% using logistic regression.

Finally, in the very recent Task 11 of SemEval 2015 (Ghosh et al., 2015), fifteen participants proposed systems to address the sentiment analysis of tweets employing figurative language (i.e. irony, sarcasm and metaphor). Those systems mainly relied on supervised learning methods (i.e. Support Vector Machines (SVMs) and regression models over carefully engineered features). The best of them for ironic and sarcastic tweets achieved respectively a precision of 0.918 (Xu et al., 2015) and 0.904 (Gimenez et al., 2015) on a test set containing 4,000 tweets.

3 Methodology

Our method is divided into two main modules as shown in Figure 1. Each module generates various kinds of features, which will be used to classify the ironic and sarcastic tweets on an 11 points scale ranging from -5 to +5. The regression tree algorithm RepTree (Thaseen and Kumar, 2013) implemented in Weka (Hall et al., 2009) is used for training and predicting the sentiment intensity of figurative data.

3.1 Data pre-processing

Before extracting the features, the tweets were preprocessed using the Stanford Lemmatizer[2] in order to transform the words in the tweets into lemmas. Then, a set of heuristic rules was created to handle the unregulated and arbitrary nature of the texts that cannot be recognized by the Stanford Lemmatizer. Words in tweets may contain repeated vowels (e.g. "loooove") or unexpected capitalization (e.g. "LOVE") to emphasize certain sentiments or emoticons. Thus, the repeated vowels are removed (e.g. from "loooove" to "love") and the capitalization is normalized (e.g. from "LOVE" to "love") to improve the lemmatization and parsing accuracy. The emphasized words are saved in a special feature bag as they are important indicators of sentiments, especially when they are in sentiment lexicons. The heavy punctuation is also handled. The use of combination of exclamation and question marks (e.g. "?!?!!") will be replaced with only a single mark (e.g. "?!"). Another step we also consider is the segmentation of the words. The segmentation is, in fact, often lost in tweets (e.g. "yeahright"). Therefore, the maximal matching algorithm is applied to segment the words (e.g. "yeah right"). In addition, all usernames, URLs and hashtags are removed from tweets as they do not provide any information about the sentiments and they might become noise for the classification process. Finally, the Stanford parser[3] was used to generate the POS tags and dependency structures of the normalized tweets.

3.2 Module 1

The overview of the module 1 is shown in Figure 1. It is based on the algorithm presented in SemEval 2015 task 11 (Xu et al., 2015). In the feature extraction sub-module, eight kinds of features are extracted.

- Token based features:
  - The "UniToken" refers to uni-grams of tokens.
  - The "BiToken" refers to bi-grams of tokens.
  - The "DepTokenPair" refers to "parent-child" pairs in the dependency structures of the tweets.
  - The "additional features" refers to the emphatic features capturing four ways twitter users express their emotions: $duplicate\_vowel$ ("loooove"), $capitalized$ ("LOVE"), $heavy\_punctuation$ ("?!?!?"), and $emoticon$ (":-D").

- Polarity dictionary based features:
  - The "PolarityWin" stores the sum of the polarity values of all the tokens in a tweet. A window size of five is used to verify whether negations are present. If a negation is present, the resulting value is set to zero. Besides, the sum of the polarity values of the tokens of the same POS tags are also stored in a different dimension. This is to measure the contributions on polarity values by different POS tags.
  - The "PolarityDep" is similar to "PolarityWin", but it differs in that the negation is checked based on the dependency structure.
  - The "PolarShiftWin" measures the difference between the most positive item and the most negative item in a window of size 5.
  - The "PolarShiftDep" measures the polarity difference of "parent-child" pairs in the dependency structures of the tweets.

Four sentiment dictionaries were used: Opinion Lexicon (Hu and Liu, 2004), Afinn (Nielsen, 2011), MPQA (Wiebe et al., 2005), and SentiWordnet (Baccianella et al., 2010). The union and intersection of the four dictionaries are also used as two additional dictionaries. Formally, the polarity feature can be represented as a (*key, val*) pair, where the key is <*pos, dict*>, or <*dict*>. For example, (<*adj, mpqa*>, 1.0) means that according to the dictionary MPQA, adjectives contribute to the polarity value

[2] http://nlp.stanford.edu/software/corenlp.shtml

[3] http://nlp.stanford.edu/software/lex-parser.shtml

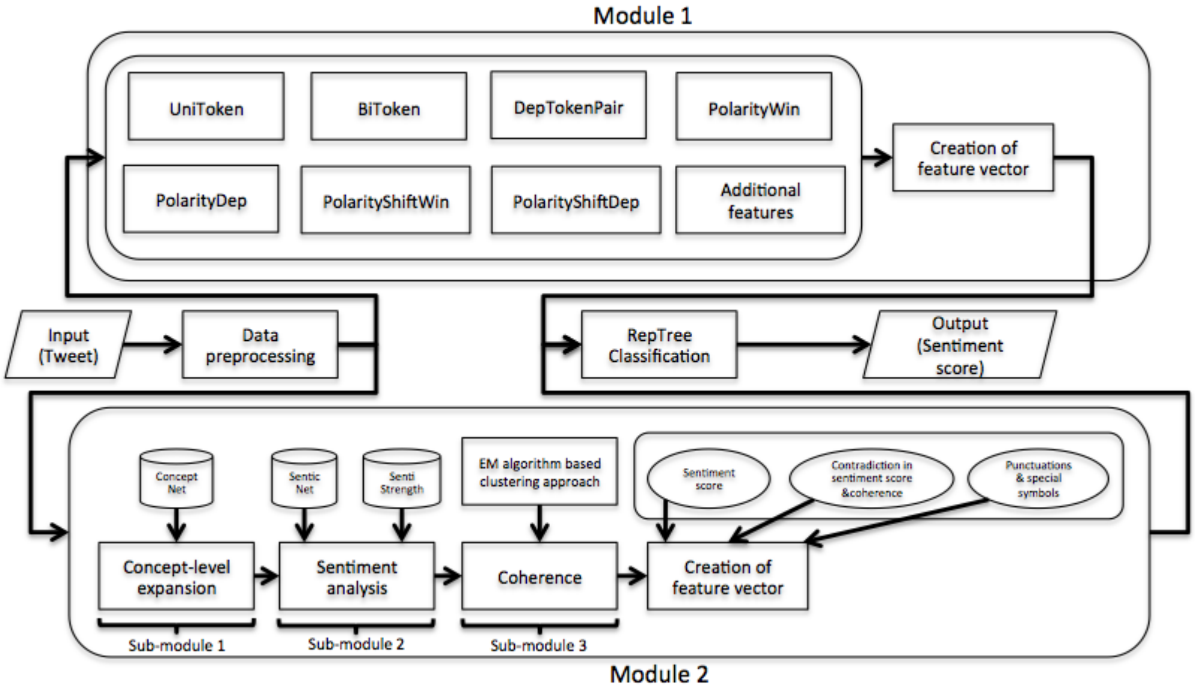

Figure 1: Flowchart of overall process of our method

for 1.0. Finally, features that occur less than three times are excluded.

In feature normalization, all the feature values are normalized within [-1, 1] based on Equation 1, where $f_{i,j}$ is the value of feature j in the ith example, and N is the sample size.

$$norm(f_{i,j}) = \frac{f_{i,j}}{\max\limits_{1 \leq k \leq N} |f_{k,j}|} \quad (1)$$

We perform feature selection through the correlative coefficient measure (Pearson's r score). A threshold value of r is used to rule out less important features. In the experiment, the correlative coefficient threshold is set to r = 0.035.

3.3 Module 2

The second module, described in Figure 1, relies on features that were proven to be effective in Tungthamthiti et al. (2014). These features include sentiment polarity score, coherence and punctuation features. Their identification and usage have been improved to become more suitable for the sentiment prediction rather than the sarcasm identification task. Note that weights of all features in the module 2 are binary.

3.3.1 Sentiment analysis

In this subpart of module 2, we create features that rely on sentiment analysis as well as the semantic analysis of tweets using concepts and commonsense knowledge.

The algorithm consists of two main steps. In the first subpart, ConceptNet[4] is used to expand the concepts for the words whose sentiment score are unknown in the SentiStrength lexicon (Cambria et al., 2010). The expanded concepts provide effective information that would benefit the task of sentiment analysis. In the second subpart of module 2, the sentiment polarity scores are calculated for each word and its expanded concepts within a tweet. Then, we create seven features. Six of them are created as an indicator of positive and negative phrases according to three possible classes (low, $medium$ and $high$). In addition, sarcasm can be recognized as a contrast between a positive sentiment referring to a negative situation (Ellen et al., 2013). Thus, another feature is created as a contradiction in sentiment score feature. This feature is activated when there exists both a positive and a negative polarity word within a tweet.

[4] http://conceptnet5.media.mit.edu

3.3.2 Coherence identification

A new method for coherence identification is proposed. As explained earlier, the contradiction of the polarity in a tweet is a useful clue. However, if positive and negative sentences mention different topics (i.e. they are incoherent), conflict of the polarity may not indicate the sarcasm or irony. Therefore, the module 2 identifies coherence in a tweet and uses it as a feature.

There are several studies related to coherence identification. A set of heuristic rules based on grammatical relations was proposed to identify coherence in tweets (Tungthamthiti et al., 2014). A more complex method, based on machine learning, was presented by Soon et al. (2001) to link coreferring noun phrases both within and across sentences. However, such method would not be appropriate for our scope, because it focuses specifically on coreference resolution, rather than identifying the coherence relationship. Nevertheless, it provides some useful insights, which can be exploited in our method.

The proposed method is based on unsupervised learning approach. Below, eleven features are created for the clustering task, in order to divide the tweets into coherence and non-coherence class. Let us suppose that sentence s_1 precedes s_2, and word w_1 and w_2 are the subject, noun or pronoun of s_1 and s_2, respectively.

1. Pronoun feature 1 – w_1 includes reflexive pronouns, personal pronouns or possessive pronouns.
2. Pronoun feature 2 – w_2 includes reflexive pronouns, personal pronouns or possessive pronouns.
3. String match feature – w_1 and w_2 are identical.
4. Definite noun phrase feature – w_2 starts with the word "the".
5. Demonstrative noun phrase feature – w_2 starts with the "this", "that", "these" and "those".
6. Both proper names feature – w_1 and w_2 are both the name entities. Two or more sentences contain proper names recognized by the Stanford Named Entity Recognizer (NER)[5].
7. Coreference resolution – two or more sentences contain coreference resolution property recognized by Stanford Deterministic Coreference Resolution System[6].
8. Semantic class agreement feature – w_1 and w_2 are semantically similar. In order to identify the word similarity, the method consists of three steps:
   - First, we create lists of synsets for both w_1 and w_2. SenseLearner 2.0[7] is used to disambiguate the meaning of the words, which allows only the suitable synsets of w_1 and w_2 to make similarity comparison.
   - Then, all possible combinations of synsets that belong to each w_1 and w_2 are compared to evaluate the similarity between them. A method proposed by Resnik (1995) is used to define the similarity between two synsets based on the information content of their lowest super-ordinate (most specific common subsumer).
   - The feature is activated when the similarity of one of synset pairs is greater than a threshold. It is set to 1.37 by our intuition.
9. Number agreement feature – w_1 and w_2 agree in number (i.e., they are both singular or plural)
10. Acronyms and abbreviation – A tweet contains an acronym or abbreviation (i.e., "lol", "ynwa").
11. Emoticons – A tweet contains an emoticon (i.e., ":-)", "☺").

After conducting a preliminary experiment, we found that the EM (expectation maximization) algorithm outperforms other approaches, including hierarchical, k-mean and DBScan, in the identification of coherence in tweets. Therefore, EM algorithm is used to cluster the tweets into two groups, one for coherent and one for non-coherent tweets. Then, a cluster label is used as the feature.

3.3.3 Punctuations and special symbols

In addition, features for punctuations and special symbols are also included in our research. The following 7 indicators are considered to determine the weights for punctuation features: number of emoticons, number of repetitive sequence of punctuations, number of repetitive sequence of characters, number of capitalized words, number of slang and booster words, number of exclamation marks and number of idioms. We use low, $medium$ and $high$ as possible scores to describe the frequency of punctuations and symbols in a tweet. These features amount to 7 × 3 = 21.

4 Experiment

In this section, we describe how the experiments were conducted to evaluate the performance of our method.

[5] http://nlp.stanford.edu/ner/

[6] http://nlp.stanford.edu/projects/coref.shtml

[7] http://web.eecs.umich.edu/~mihalcea/downloads.html#senselearner

4.1 Data

In our experiment, we used the training and test data distributed for SemEval-2015 Task 11 on "Sentiment Analysis of Figurative Language in Twitter"[8]. The data set consists of tweets containing sarcasm, irony, metaphor and non-figurative tweets. The training set contains 7,952 tweets, while the test set contains 4,000 tweets. All tweets are manually annotated with a fine-grained sentiment scale value in 11 points (between -5 to +5).

4.2 Task

The task is to estimate a degree of fine-grained sentiment score for each tweet in the dataset. There are two subtasks. One is to predict the sentiment score by 5-fold cross validation on the training set (reported in Subsection 5.1). In this task, the effectiveness of individual features is mainly investigated. The other is to predict the sentiment intensity of the test set using the model learned from the training data (reported in Subsection 5.2). The performance of the proposed method is analyzed considering several types of tweets (sarcastic, ironic, metaphorical and non-figurative ones).

4.3 Baselines

In this study, two baselines are created. One was developed as a naive prediction using the average polarity value of the training data, while the other one uses supervised machine learning (RepTree) with UniToken (uni-gram) features to train classifier for sentiment classification.

4.4 Evaluation measures

Cosine similarity and root mean squared error (RMSE) are used as the evaluation criteria of sentiment intensity estimation. They illustrate how similar the predicted values and the actual annotated values are. They can be calculated by using equation 2 and 3, respectively.

$$Cosine[a,b] = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2} \times \sqrt{\sum_{i=1}^{n}(b_i)^2}} \qquad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a-b)^2} \qquad (3)$$

- i refers to the value of tweet index.
- n refers to the number of tweets.
- a refers to the human-annotated sentiment score of tweet i.
- b refers to the predicted sentiment score of tweet i by our system.

5 Results and discussion

5.1 Results on the training data

Table 1: Results of the module 1 of 5-fold cross validation on the training data

Method	Cosine	RMSE
Avg. polarity (Baseline1)	0.818	1.985
UniToken (Baseline 2)	0.851	1.682
UniToken		
+BiToken	0.849	1.700
+DepTokenPair	0.851	1.673
+PolarityWin	0.852	1.657
+PolarityDep	0.854	1.643
+PolarityShiftWin	0.854	1.640
+PolarityShiftDep	0.854	1.640

Table 1 shows the results of the two baselines and those of module 1, trained with UniToken and one additional feature on the training data. Surprisingly, the average polarity value (baseline 1) and the classification based on UniToken features (baseline 2) were powerful predictor of the sentiment. Both methods achieved relatively high cosine values (i.e. 0.818 and 0.851, respectively). In particular, it is interesting to notice that baseline 1 can achieve such results because the majority of the tweets are annotated with moderate negative values, varying from -2 to -3. Accordingly, the average polarity value of words computed by our baseline system also indicates the moderate negative range. Thus, baseline 1 achieved a high accuracy and also became competitive with other methods.

BiToken and DepTokenPair. As can be seen, the result shows that all features have taken part in the method to enhance the accuracy, except for BiToken. Thus, we can easily conclude that BiToken is not a relevant feature for sentiment prediction of figurative tweets.

PolarityWin and PolarityDep features. The features contributed some improvements to the overall result. The reason is that these features handle the negations, which often occurs within the figurative tweets.

PolarityShiftWin and PolarityShiftDep features. The result also indicates that PolarityShiftWin and

[8] http://alt.qcri.org/semeval2015/task11/

PolarityShiftDep features contributed to some improvement towards the overall result. The difference between the most positive and negative items can represent the strength of the overall polarity and also indicate if there exists a conflict in a tweet, which may reveal either irony or sarcasm. As a result, we can conclude that the shift in polarity value has an impact on the sentiment prediction for figurative tweets.

Table 2: Results of the module 2 of 5-fold cross validation on the training data

Method	Cosine	RMSE
All features (module 2)	0.825	1.376
– Sentiment contradiction	0.821	1.384
– Sentiment score	0.803	1.511
– Punctuations + symbols	0.820	1.402
– Coherence	0.817	1.425
– Concept level knowledge	0.781	1.658

Table 2 shows the overall result of the module 2 and also how the results change as the features are removed. Cosine value and RMSE of the module 2 were 0.825 and 1.376, which were better than baseline 1 but worse than baseline 2.

Punctuations and special symbols. The feature contributed to some improvement to the overall method. The cosine value is reduced by 0.005 (from 0.825 to 0.803) when the feature is removed. Figurative tweets often contain emoticons and heavy punctuation marks to simulate the gestural signs, onomatopoeic expressions and also boosting the intensity of emotion. Therefore, the feature can be used to capture this particular characteristic.

The concept-level knowledge. Expansion of the concepts implemented in the first subpart can also enhance the performance of the sentiment score. Tweets are considered as unstructured and context free data. There are many words and slangs, which cannot be compiled in any dictionaries. Concept-level and common-sense knowledge are applied to compensate to such lack with related concepts, which allows the system to compute the sentiment score more accurately.

Coherence identification. In our experiment, it is clearly shown that coherence feature has an impact on the improvement of the result. This is a proof that it is necessary to verify whether there are terms referring to each other across the sentences, in order to make the contradiction identification more effective.

Table 3: Results of the integrated system of 5-fold cross validation on the training data

Method	Cosine	RMSE
Module 1	0.859	1.256
Module 2	0.825	1.376
Integrated module 1 & 2	0.882	1.154

Table 3 shows the results comparison of the module 1, module 2 and integration of them. The results show that the integration of the module 1 and 2 performs significantly better than the baseline 2 that uses uni-gram feature. It is also clearly shown that the cosine value of the integrated system outperforms each module 1 and 2 by 0.023 and 0.057, respectively.

5.2 Results on the test data

Table 4: Results of the module 1 on the test dataset

Category	Cosine	RMSE
Sarcasm	0.896	0.997
Irony	0.918	0.671
Metaphor	0.535	3.917
Non-figurative	0.290	4.617
Overall	0.687	2.602

Table 5: Results of the module 2 on the test dataset

Category	Cosine	RMSE
Sarcasm	0.948	0.732
Irony	0.912	0.851
Metaphor	0.389	4.165
Non-figurative	0.207	4.682
Overall	0.542	2.030

Table 4 shows the results of sentiment prediction of the module 1 on the test data. The performance is effective on sarcastic and ironic data, since the module 1 achieved the cosine value of 0.896 and 0.918, respectively. However, the performance is rather poor when we attempted to predict the sentiment score for metaphor and non-figurative tweets. In Table 5, the results of the module 2 seem to be very competitive to the module 1 in all categories. The cosine value was higher for sarcasm tweets and comparable for irony tweets. The major differences in the module 1 and 2 are the use of the concept expansion and coherence feature. They seem especially work well for guessing the sentiment score of the sarcasm tweets.

Table 6: Results of the integrated system on the test dataset

Category	**Cosine**	**RMSE**
Sarcasm	0.953	0.718
Irony	0.921	0.821
Metaphor	0.561	3.899
Non-figurative	0.297	4.520
Overall	0.736	1.382

Table 6 shows the results of the integrated system, clearly indicating that the overall result of the proposed method is much better than both the module 1 and 2. Thus, it is obvious that the feature sets of both modules complement each other when they are integrated into a single method. Table 7 shows the comparison of the cosine measure among our system and the five top systems participated in SemEval 2015 Task 11. Note that our system largely outperformed all the other 15 participating systems on the ironic and sarcastic tweets, although achieved second in the overall dataset.

Table 7: Comparison of the our result against five top peer systems participated in SemEval 2015 Task 11

System	All	S	I	M	N
ClaC	**0.758**	0.892	0.904	**0.655**	**0.584**
UPF	0.711	0.903	0.873	0.520	0.486
LLT_PolyU	0.687	0.896	0.918	0.535	0.290
LT3	0.658	0.891	0.897	0.443	0.346
elirf	0.658	0.904	0.905	0.411	0.247
Our system	0.736	**0.953**	**0.921**	0.561	0.297

Note: S = sarcasm, I = irony, M = metaphor, N = non-figurative
ClaC = Concordia university; UPF = Universitat Pompeu Fabra; LLT_PolyU = Hong Kong Polytechnic University; LT3 = Ghent University; elirf = Universitat Politecnica de Valencia

The performance of our system as well as the participating systems in SemEval 2015 was much better for the sarcasm and irony than metaphor and non-figurative. It may be worthy noticing here that most of the mentioned models were developed keeping in mind that sarcasm and irony mostly rely on incongruity (i.e. logical inconsistency), while metaphor and non-figurative texts rely on congruity[9]. Therefore, the systems designed to identify incongruity poorly perform on the congruous texts. It suggests that the sarcasm/irony and metaphor/non-figurative are needed to be handled differently.

5.3 Paired t-Test

Table 8: Paired t-test results between the module 1 or the module 2 and the integration of module 1 and 2

Pair 1: Module 1 - integrated modules 1 & 2	
$P(T <= t)$ one-tail	0.069
$P(T <= t)$ two-tail	0.098
Pair 2: Module 2 - integrated modules 1 & 2	
$P(T <= t)$ one-tail	0.029
$P(T <= t)$ two-tail	0.047

A paired t-test was conducted to see whether there was a statistical significant difference between the module 1 or module 2 and the integration of them. Table 8 shows two results of paired t-test: ‘pair 1’ between the module 1 and the integrated system, and ‘pair 2’ between the module 2 and the integrated system. α value was 0.029 (one-tail) and 0.047 (two-tail) for the pair 1 and also 0.069 and 0.098 for the pair 2. Since the α values of both pairs are less than 0.1, we can conclude that there was a significant difference in the mean scores between both pairs with 90% confident interval.

6 Conclusion

In this research, we present a model for the prediction of fine-grained sentiment score for sarcastic and ironic tweets. The method consists of two modules that are refined from the previous methods, also introducing some new features. The results of the experiments indicate that our proposed method is better than the strong baselines, and integration of two modules achieves the best result among the participating systems in SemEval-2015 for the sarcastic and ironic tweets. On top of the features derived from two previous well performing systems, we enriched the feature set with several new implemented ones. In particular, the “additional features” is added to the module 1, while the counters of several punctuations & special symbols and a new method to identify “coherence feature” is proposed in the module 2. The contribution of each feature has been carefully analyzed and reported.

In the near future, we intend to apply the feature set to different tasks. One of them is to predict whether a tweet contains irony or sarcasm, rather than calculating the sentiment score. Other applications will be explored.

[9] In metaphor, a concept in a target domain is expressed by terms from a source domain, but there is no incongruity among the used terms and concepts.

References

Baccianella S., Esuli A., and Sebastiani F.. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Barbieri F. and Saggion H. 2014, Modelling irony in twitter, *In Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 56–64

Buschmeier K., Cimiano P., and Klinger R. 2014, An impact analysis of features in a classification approach to irony detection in product reviews

Cambria E., Speer R., Havasi C. and Hussain A. 2010, SenticNet: A Publicly Available Semantic Resource for Opinion Mining, *Commonsense Knowledge: Papers from the AAAI Fall Symposium*

Carvalho P., Sarmento L., Silva J. M. and Oliveira D. E. 2009, Clues for detecting irony in user-generated contents: oh...!! its so easy;-), *In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56

Davidov D., Tsur O., and Rappoport A. 2010, Semi-supervised recognition of sarcastic sentences in twitter and amazon, *In Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Association for Computational Linguistics

Ellen R., Ashequl Q., Prafulla S., Lalindra S., Gilbert D. S., Gilbert N., Ruihong H. 2013, Sarcasm as Contrast between a Positive Sentiment and Negative Situation, *In Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 704–714, Seattle, Washington

Finn Årup Nielsen. 2011, A new anew: Evaluation of a word list for sentiment analysis in microblogs, In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages.*

Ghosh A., Li G., Veale T., Rosso P., Shutova E., Reyes A., and Barnden J. 2015, Semeval-2015 task 11: Sentiment analysis of figurative language in twitter, *In Proceedings of the International Workshop on Semantic Evaluation (SemEval-2015), Co-located with NAACL and *SEM*, Denver, Colorado, USA

Gimenez M., Pla F. and Hurtado L.F. 2015, ELiRF: A Support Vector Machine Approach for Sentiment Analysis Tasks in Twitter at SemEval-2015, *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Association for Computational Linguistics, pages 673–678, Denver, Colorado

Grice H. P. 1975, Logic and conversion, *Syntax and semantics 3: Speech arts*, pages: 41–58

Haiman J. 1998, Talk is cheap: Sarcasm, alienation, and the evolution of language, *Language Arts & Disciplines*, Oxford, Oxford University Press

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I.H. 2009, The weka data mining software: An update, *SIGKDD Explorations*

Hao Y. and Veale T. 2010, An ironic fist in a velvet glove: Creative mis-representation in the construction of ironic similes, *Minds and Machines*, 20(4):635–650

Hart L. 2013, The Linguistics of Sentiment Analysis, Portland State University, PDX Scholar 2013, http://pdxscholar.library.pdx.edu/honorstheses/20

Martin J. R. and White P. R.R. 2005, The Language of Evaluation: Appraisal in English, *Palgrave*, London, UK

Miller A. G. 1995 WordNet: A Lexical Database for English *Communications of the ACM*

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

Pak E. and Paroubek P. 2010, Twitter as a corpus for sentiment analysis and opinion mining, *In Proceedings of the Seventh Conference on International Language Resources and Evaluation*

Pang B. and Lee L. 2008, Opinion mining and sentiment analysis, *Foundations and Trends in Information Retrieval* Vol. 2, pages 1–135

Quintilian 1953, The institutio Oratoria of Quintilian. With an English Translation by Harold Edgeworth Butler. London: William Heinemann

Resnik P. 1995, Using Information Content to Evaluate Semantic Similarity in a Taxonomy, *CoRR*

Reyes A., Rosso P. 2012, Making objective decisions from subjective data: detecting irony in customer reviews, *Decision Support System 2012*, 53:754–760.

Reyes A., Rosso P., and Veale T. 2013, A multidimensional approach for detecting irony in twitter, *Language Resources and Evaluation*, 47(1):239–268.

Soon W. M., Ng H. T, Lim D. C. Y. 2001 A Machine Learning Approach to Coreference Resolution of Noun Phrases *Computational Linguistics*, pages 521–544, Cambridge, MA, USA

Stevenson A. 2010, Oxford dictionary of English, Oxford University Press

Stringfellow, F. J. 1994. *The Meaning of Irony* NewYork: State University of NY.

Thaseen S. and Kumar C. A. 2013, An analysis of supervised tree based classifiers for intrusion detection system, *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*, pages 294–299, Salem, MA, USA

Tsur O., Davidov D. 2010 ICWSM - a great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews *In International AAAI Conference on Weblogs and Social*

Tungthamthiti P., Kiyoaki S., and Masnizah M. 2014, Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches, *In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, pages 404–413, Phuket, Thailand

Wiebe J., Wilson T., and Cardie C. 2005. Annotating expressions of opinions and emotions in language, *Language resources and evaluation*, 39(2-3):165–210.

Xu, Hongzhi and Santus, Enrico and Laszlo, Anna and Huang, Chu-Ren 2015, LLT-PolyU: Identifying Sentiment Intensity in Ironic Tweets, *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Association for Computational Linguistics, pages 673–678, Denver, Colorado