Dependency Tree-based Rules for Concept-Level Aspect-based Sentiment Analysis

Soujanya Poria[1,4], Nir Ofek[2], Alexander Gelbukh[3], Amir Hussain[4], and Lior Rokach[2]

[1] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
[2] Department of Information Systems Engineering, Ben Gurion University, Israel
[3] Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico
[4] Department of Computing Science and Mathematics, University of Stirling, UK
sporia@ntu.edu.sg,soujanya.poria@cs.stir.ac.uk,
nirofek@bgu.ac.il,www.gelbukh.com,ahu@cs.stir.ac.uk,liorrk@bgu.ac.il

Abstract. Over the last few years, the way people express their opinions has changed dramatically with the progress of social networks, web communities, blogs, wikis, and other online collaborative media. Now, people buy a product and express their opinion in social media so that other people can acquire knowledge about that product before they proceed to buy it. On the other hand, for the companies it has become necessary to keep track of the public opinions on their products to achieve customer satisfaction. Therefore, nowadays opinion mining is a routine task for every company for developing a widely acceptable product or providing satisfactory service. Concept-based opinion mining is a new area of research. The key parts of this research involve extraction of concepts from the text, determining product aspects, and identifying sentiment associated with these aspects. In this paper, we address each one of these tasks using a novel approach that takes text as input and use dependency parse tree-based rules to extract concepts and aspects and identify the associated sentiment. On the benchmark datasets, our method outperforms all existing state-of-the-art systems.

1 Introduction

For each sentence, our system generates a list of aspects (Section 3) and the polarity of the sentiment expressed in the sentence (Section 4). Our sentiment analysis process relies on extraction of concepts (Section 2). Online demos of concept and aspect parsing and the sentiment analysis are available on `http://www.sentic.net/demo`.

2 Concept Parser

Concept parsing is crucial for such tasks as concept-based opinion mining [1,2], big social data analysis [3], and crowd validation [4]. The proposed concept parser deconstructs input sentences into multi-word expressions. For this, it extracts concepts from the dependency parse tree of the sentence[5] basing on hand-crafted rules. Examples of such rules are given below; see more details in [5].

[5] We used the Stanford Dependency parser, `http://nlp.stanford.edu/software/lex-parser.shtml`.

Subject Noun Rule If the active token *h* is in a subject noun relationship with a verb *t*, then the concept *t-h* is extracted. E.g., in (1), *movie* is the subject of *boring*; the concept `boring-movie` is extracted.

(1) The movie is boring.

Joint Subject Noun and Adjective Complement Rule If the active token *h* is in a subject noun relationship with a verb *t* and *t* is in adjective complement relationship with an adverb *w*, then the concept *w-h* is extracted. E.g., in (2), *flower* is the subject of *smells*, which is in adjective complement relationship with *bad*; the concept `bad-flower` is extracted.

(2) The flower smells bad.

Experiments and Results To calculate the performance, we selected 300 sentences from the *Stanford Sentiment Dataset* [6] and extracted the concepts manually, which gave 3204 concepts. On these sentences, our parser achieved 92.01% accuracy.

3 Aspect Parser

Aspect-based opinion mining aims to model relations between the polarity of a document and its opinion targets, or aspects. Our system is able to extract both implicit and explicit aspects; see [7] for more details on our aspect parser.

Compilation of an implicit aspect lexicon We used the product review dataset described in [8,9] to create the implicit aspect lexicon. We selected from the dataset the sentences that had implicit aspects. From those sentences, we extracted the implicit aspect clues and manually labeled them with suitable categories. For example, from the sentence *The car is expensive* we extracted the implicit aspect clue *expensive* and labeled it with the category *price*. We identified in this corpus the following categories: *functionality*, *weight*, *price*, *appearance*, *behavior*, *performance*, *quality*, *service*, and *size*. For each identified implicit aspect clue, we also retrieved its synonyms from WordNet. This gave us a lexicon of 1128 implicit aspect clues labeled by aspect categories listed above.

Opinion Lexicon We used SenticNet 3.0 [10,11,12,13] as an opinion lexicon. It contains 14,000 common sense-knowledge concepts labeled by their polarity scores.

Algorithm We used the Stanford Dependency parser to obtain the dependency tree of each sentence. Then we employed a complex system of hand-crafted rules on these parse trees to extract the aspects. Examples of such rules are given below. Some of our rules block the application of other rules, so the rules given below are not always applied.

Subject Noun Rule If the active token *h* is in a subject noun relationship with a word *t*, then:

1. If *t* has an adverbial or adjective modifier that exists in the SenticNet, then we extract *t* as an aspect. E.g., in (3), according to Stanford parser, *it* is in a subject noun relationship with *camera*, which has an adjective modifier *nice*, so *camera* is extracted.

   (3) It is a nice camera.

2. If the sentence has no auxiliary verb (*is*, *was*, *would*, *should*, *could*, etc.), then:
   – If *t* is a verb modified by an adjective or adverb or it is in *adverbial clause modifier* relation with another token, then both *h* and *t* are extracted as aspects. E.g., in (4), *battery* is in a subject relation with *lasts*, so the aspects *last* and *battery* are extracted.

     (4) The battery lasts little.

   – If *t* has a noun *n* as a direct object, *n* is in SenticNet, and *n* is in a prepositional relation with another noun *m*, then both *n* and *m* are extracted as aspects. E.g., in (5), *like* is in direct object relation with *beauty*, which is connected to *screen* via a preposition relation. So the aspects *screen* and *beauty* are extracted.

     (5) I like the beauty of the screen.

3. Copula is the relation between the complement of a copular verb and the copular verb. If the token *h* existing in the implicit aspect lexicon is in a copula relation with a couplar verb, then we extract *h* as an aspect. E.g., in (6) *expensive* is extracted as an aspect.

   (6) The car is expensive.

Sentences with no subject noun relation in the parse tree We extracted the aspects from such sentences using the following rules:

1. If an adjective or adverb *h* is in infinitival or open clausal complement relation with a token *t* and *h* exists in the implicit aspect lexicon, then we extract *h* as an aspect. E.g., in (7) we extract *big* as an aspect, since it is connected to *hold* via a clausal complement relation.

   (7) Very big to hold.

2. If a token *h* is connected to a noun *t* via a prepositional relation, then we extract both *h* and *t* as aspects. E.g., in (8), *sleekness* is extracted as an aspect.

   (8) Love the sleekness of the player.

Obtaining implicit aspect categories After obtaining the aspects using these rules, we retrieved the categories of the implicit aspects from the implicit aspect lexicon.

Experiments and Results We experimented on the Semeval 2014 aspect-based sentiment analysis data.[6] On this dataset, we obtained 91.25% precision and 88.12% recall.

4 Common Sense Knowledge-based Sentiment Analysis

This section describes the algorithm we used to compute the polarity score of a sentence. We have introduced a novel paradigm for concept-level sentiment analysis that merges linguistics, common-sense computing, and machine learning for improving the accuracy of tasks such as polarity detection [14]. By allowing sentiments to flow from concept to concept based on the dependency relation of the input sentence, in particular, we achieve a better understanding of the contextual role of each concept within the sentence. With this, our polarity detection engine outperforms the state-of-the-art statistical methods. Below we describe some rules we used and the ensemble classification process; see [14] for more details on our concept-level sentiment analysis algorithm.

Dependency Rules We used rules based on specific dependency patterns to drive the way concepts were searched in SenticNet. Below are some examples of such rules.

Subject nouns This rule is applied when the active token *h* is the syntactic subject of a word *t*. If the complex concept *t-h* was found in SenticNet, then it was used to calculate the polarity of the relation (otherwise, other rules are activated later). E.g., in (9), *movie* is in a subject relation with *boring* and `(boring-movie)` is in SenticNet, so its corresponding polarity was used.

(9) The movie is boring.

Adjective and clausal complements These rules deal with verbs having as complements either an adjective or a closed clause (i.e. a clause, usually finite, with its own subject).

1. If the active token is head verb of one of the complement relations, then first the algorithm looks for the binary concept *h d*. If it is found, the relation inherits its polarity properties. If it is not found:
   - if both elements *h* and *d* are independently found in SenticNet, then we take sentiment of the *d* as the sentiment of the relation.
   - if the dependent *d* alone is found in SenticNet, its polarity is attributed to the relation

   E.g., in (10), *smells* is the head of a dependency relation with *bad* as the dependent; the relation inherits the polarity of *bad*.

   (10) This meal smells bad.

2. If the active token is modified by a relative clause, restrictive or not, and the dependent is the verb of the relative clause (usually it is), then if the binary concept *h d* is found in SenticNet, then it assigns polarity to the relation, otherwise the polarity is assigned (in order of preference):

[6] `http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools`

– By the value of the dependent verb *d* if it can be found;
– By the value of the active token *h* if it is found in SenticNet.

E.g., in (11) *movie* is in relation with *love* which acts as a modifier in the relative clause.

(11) I saw the movie you love.

Assuming *love movie* is not in SenticNet and that *love* is, then the latter will contribute the polarity score of the relation. If neither of these two is in SenticNet, then the dependency will receive the score associated with *movie*.

Machine Learning Technique For each sentence, we extracted concepts from it as explained in Section 2 and looked them up in SenticNet. If we found at least one concept in SenticNet, then we used our knowledge-based method to detect sentiment. Otherwise, we resorted to our machine learning-based technique. The Machine Learning module was trained on the Blitzer dataset. Below we describe some of the features we used for training.

Sentic feature The polarity scores of each concept extracted from the sentence were obtained from the SenticNet and summed up to produce a single scalar feature. This feature was used for training, but was not available in testing.

Part-of-speech features This feature was defined by the total numbers of adjectives, adverbs, and nouns in the sentence, which gave three distinct features.

Modification feature This binary feature was set to 1 if we found any modification relation in the sentence; otherwise it was set to 0.

Results on the Blitzer-derived Dataset At the sentence level, on the Blitzer dataset 87.00% accuracy was achieved.

5 Conclusion

We gave examples of dependency tree-based rules to extract concepts, aspects, and sentiment polarity from natural language texts. In our future work, we aim to extend this work by adding more rules or using existing rules to find new rules based on association-based rules. We also aim to use extend the algorithm so that it can be applied to other languages. Techniques based on recognizing textual entailment [15,16] will help us to achieve multilingual application of our algorithm. For aspect extraction, we will explore the role of adjectives [17] which are often used to modify the aspects in the opinionated text. Thus, identifying the role of adjectives will help us to extract the aspects from the text with a higher precision.

References

1. Poria, S., Gelbukh, A., Hussain, A., Howard, N., Das, D., Bandyopadhyay, S.: Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intelligent Systems **28**(2) (2013) 31–38
2. Poria, S., Gelbukh, A., Cambria, E., Hussain, A., Huang, G.B.: EmoSenticSpace: A novel framework for affective common-sense reasoning. Knowledge-Based Systems (2014)
3. Cambria, E., Rajagopal, D., Olsher, D., Das, D.: Big social data analysis. Big Data Computing (2013) 401–414
4. Cambria, E., Hussain, A., Havasi, C., Eckl, C., Munro, J.: Towards crowd validation of the UK national health service. WebSci10 (2010)
5. Poria, S., Agarwal, B., Gelbukh, A., Hussain, A., Howard, N.: Dependency-based semantic parsing for concept-level text analysis. In: Computational Linguistics and Intelligent Text Processing. Proc. 15th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014, Part I. Volume 8403 of Lecture Notes in Computer Science., Springer (2014) 113–127
6. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). (2013)
7. Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Workshop Proceedings of the The 25th International Conference on Computational Linguistics, COLING 2014. COLING 2014 Org. Com. (2014)
8. Cruz-Garcia, I.O., Gelbukh, A., Sidorov, G.: Implicit aspect indicator extraction for aspect-based opinion mining. Submitted (2014)
9. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Computational linguistics **37**(1) (2011) 9–27
10. Cambria, E., Olsher, D., Rajagopal, D.: Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. AAAI (2014)
11. Poria, S., Gelbukh, A., Cambria, E., Yang, P., Hussain, A., Durrani, T.: Merging SenticNet and WordNet-Affect emotion lists for sentiment analysis. In: 11th International Conference on Signal Processing (ICSP 2012). Volume 2., IEEE (2012) 1251–1255
12. Poria, S., Gelbukh, A., Cambria, E., Das, D., Bandyopadhyay, S.: Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In: 12th International Conference on Data Mining Workshops (ICDMW 2012), IEEE (2012) 709–716
13. Poria, S., Gelbukh, A., Das, D., Bandyopadhyay, S.: Fuzzy clustering for semi-supervised learning—case study: Construction of an emotion lexicon. In: Advances in Computational Intelligence. Proceedings of MICAI 2012. Volume 7629 of Lecture Notes in Artificial Intelligence., Springer (2013) 73–86
14. Poria, S., Cambria, E., Winterstein, G., Huang, G.B.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems, DOI: 10.1016/j.knosys.2014.05.005 (2014)
15. Pakray, P., Pal, S., Poria, S., Bandyopadhyay, S., Gelbukh, A.: JU_CSE_TAC: Textual entailment recognition system at TAC RTE-6. In: System Report, Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook. (2010)
16. Pakray, P., Neogi, S., Bhaskar, P., Poria, S., Bandyopadhyay, S., Gelbukh, A.: A textual entailment system using anaphora resolution. In: System Report. Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook. (2011)
17. Ofek, N., Rokach, L., Mitra, P.: Methodology for connecting nouns to their modifying adjectives. In: Computational Linguistics and Intelligent Text Processing. Proc. 15th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014, Part I. Volume 8403 of Lecture Notes in Computer Science. Springer (2014) 271–284