Dependency-Based Semantic Parsing for Concept-Level Text Analysis

Soujanya Poria[1,4,6], Basant Agarwal[2], Alexander Gelbukh[3], Amir Hussain[4], and Newton Howard[5]

[1] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
[2] Department of Computer Engineering, Malaviya National Institute of Technology, India
[3] Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico
[4] Department of Computing Science and Mathematics, University of Stirling, UK
[5] MIT Media Laberotory, MIT, USA
[6] The Brain Sciences Foundation, Cambridge, MA 02139, USA
sporia@ntu.edu.sg, sp47@cs.stir.ac.uk, basant@mnit.ac.in, www.gelbukh.com, ahu@cs.stir.ac.uk, nhmit@mit.edu

Abstract. Concept-level text analysis is superior to word-level analysis as it preserves the semantics associated with multi-word expressions. It offers a better understanding of text and helps to significantly increase the accuracy of many text mining tasks. Concept extraction from text is a key step in concept-level text analysis. In this paper, we propose a ConceptNet-based semantic parser that deconstructs natural language text into concepts based on the dependency relation between clauses. Our approach is domain-independent and is able to extract concepts from heterogeneous text. Through this parsing technique, 92.21% accuracy was obtained on a dataset of 3,204 concepts. We also show experimental results on three different text analysis tasks, on which the proposed framework outperformed state-of-the-art parsing techniques.

1 Introduction

Concept-level text analysis [24,26,25] focuses on a semantic analysis of text [12] through the use of web ontologies or semantic networks, which allow the aggregation of conceptual and affective information associated with natural language opinions. By relying on large semantic knowledge bases, such approaches step away from blind use of keywords and word co-occurrence count, but rather rely on the implicit features associated with natural language concepts. Unlike purely syntactical techniques, concept-based approaches are able to detect also sentiments that are expressed in a subtle manner, e.g., through the analysis of concepts that do not explicitly convey any emotion, but which are implicitly linked to other concepts that do so. The bag-of-concepts model can represent semantics associated with natural language much better than bags-of-words [4]. In the bag-of-words model, in fact, a concept such as `cloud computing` would be split into two separate words, disrupting the semantics of the input sentence (in which, for example, the word `cloud` could wrongly activate concepts related to `weather`).

The analysis at concept-level allows for the inference of semantic and affective information associated with natural language opinions and, hence, enables a comparative fine-grained feature-based sentiment analysis. Rather than gathering isolated opinions about a whole item (e.g., iPhone5), users are generally more interested in comparing different products according to their specific features (e.g., iPhone5's vs Galaxy S3's touchscreen), or even sub-features (e.g., fragility of iPhone5's vs Galaxy S3's touchscreen). In this context, the construction of comprehensive common and common-sense knowledge bases is key for feature-spotting and polarity detection, respectively. Common-sense, in particular, is necessary to properly deconstruct natural language text into sentiments— for example, to appraise the concept `small room` as negative for a hotel review and `small queue` as positive for a post office, or the concept `go read the book` as positive for a book review but negative for a movie review [2]. Common-sense knowledge describes basic understandings that people acquire through experience. In cognitive science, building conceptual representations is a fundamental ability to understand and handle objects and actors of an operating environment [15].

To this end, the proposed concept parser aims to break text into clauses and, hence, deconstruct such clauses into concepts, to be later fed to a vector space of common-sense knowledge. For applications in fields such as real-time human-computer interaction and big social data analysis, in fact, deep natural language understanding is not strictly required: a sense of the semantics associated with text and some extra information (e.g., affect) associated with such semantics are often enough to quickly perform tasks such as emotion recognition and polarity detection. Common-sense reasoning is often performed through common-sense ontologies and the employment of reasoning algorithms, such as predicate logic and machine learning, to reach a conclusion.

In this paper, we propose a novel concept parser based on the semantic relationship between words in natural language text and on the semantics of the ConceptNet ontology. The paper is organized as follows: Section 2 describes related works in semantic parsing; Section 3 discusses the proposed algorithm; Section 4 offers a summary of the novelty of our work; Section 5 presents experimental results and a comparative evaluation against the state of the art; Section 6 proposes three possible applications of the proposed concept parser; finally, Section 7 concludes the paper.

2 Related Work

Automatic knowledge mining from text is a popular research field and concept extraction is one of its key steps. [5] used domain specific ontologies to acquire knowledge from text. Using such ontologies the authors extracted 1.1 million common-sense knowledge assertions. Concept mining is useful for tasks such as information retrieval [29], opinion mining [3], text classification [35].

State-of-the-art approaches mainly exploit term extraction methods to obtain concepts from text. The approaches can be classified into two main categories: linguistic rules [7] and statistical approaches [36] [1]. [36] used term frequency and location of the words and, hence, employed a non-linear function to calculate term weighting. [1] mined concepts from the Web by using webpages to construct topic signatures of concepts and, hence, built hierarchical clusters of such concepts (word senses) that lexicalize a given word. [9] and [34] combined linguistic rules and statistical approaches to enhance the concept extraction process.

Other relevant works in concept mining focus on concept extraction from documents. Gelfand et al. have developed a method based on the Semantic Relation Graph to extract concepts from a whole document [10]. They used the relationship between words, extracted on a lexical database, to form concepts. Our approach also exploit the relationship between words but it obtain the semantic relationship between words based on dependency parsing. We gather more conceptual information of a concept using the ConceptNet ontology. Concepts extracted from text are sent as a query to ConceptNet to extract their semantics.

Nakata has described a method to index important concepts described in a collection of documents belonging to a group for sharing them [20].

Lexicon syntactic patterns is also one of the popular techniques for concept extraction. [14] extracted hyponomy relations from text from Grolier's Encyclopedia by matching 4 given lexicon-syntactic patterns. Her theory explored a new direction in the concept mining field. She claimed existing hyponomy relations can be used to extract new lexical syntactic patterns. [17] and [18] used the "isa" pattern to extract Chinese hyponymy relations from unstructured Web corpus and obtained promising results.

2.1 Part Of Speech Based Concept Parsing Model

Rajagopal et al. 2013 [28] proposed a novel Part Of Speech based approach to extract concepts. This is the only state of the art approach which tried to understand the meaning of the text. Later, we compare our approach with [28]. Below, we briefly present the POS algorithm proposed in [28].

First, the semantic parser breaks text into clauses. Each verb and its associated noun phrase are considered in turn, and one or more concepts is extracted from these. As an example, the clause "I went for a walk in the park", would contain the concepts `go walk` and `go park`. The Stanford Chunker [8] is used to chunk the input text. A sentence "I am going to the market to buy vegetables and some fruits" would be broken into "I am going to the market" and "to buy vegetables and some fruits". A general assumption during clause separation is that, if a piece of text contains a preposition or subordinating conjunction, the words preceding these function words are interpreted not as events but as objects.

The next step of the algorithm then separates clauses into verb and noun chunks, as suggested by the parse trees shown in Fig. 1. Next, clauses are normalized in two stages. First, each *verb* chunk is normalized using the Lancaster stemming algorithm [21]. Second, each potential *noun* chunk associated with individual verb chunks is paired with the stemmed verb in order to detect multi-word expressions of the form 'verb plus object'. Objects alone, however, can also represent a common-sense concept. To detect such expressions, a POS-based bigram algorithm checks noun phrases for stopwords and adjectives. In particular, noun phrases are first split into bigrams and then processed through POS patterns, as shown in Algorithm 1.

POS pairs are taken into account as follows:

1. ADJ + NOUN : The adj+noun combination and noun as a stand-alone concept are added to the objects list.
2. ADJ + STOPWORD : The entire bigram is discarded.

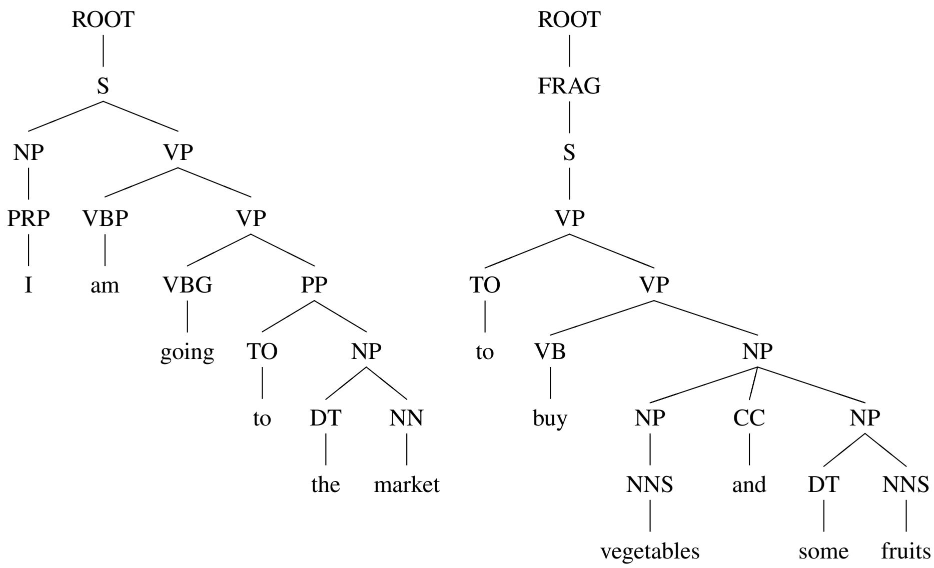

Fig. 1. Example parse trees.

3. NOUN + ADJ : As trailing adjectives do not tend to carry sufficient information, the adjective is discarded and only the noun is added as a valid concept.
4. NOUN + NOUN : When two nouns occur in sequence, they are considered to be part of a single concept. Examples include *butter scotch*, *ice cream*, *cream biscuit*, and so on.
5. NOUN + STOPWORD : The stopword is discarded, and only the noun is considered valid.
6. STOPWORD + ADJ: The entire bigram is discarded.
7. STOPWORD + NOUN : In bigrams matching this pattern, the stopword is discarded and the noun alone qualifies as a valid concept.

The POS-based bigram algorithm extracts concepts such as *market*, *some fruits*, *fruits*, and *vegetables*. In order to capture event concepts, matches between the object concepts and the normalized verb chunks are searched. This is done by exploiting a parse graph that maps all the multi-word expressions contained in the knowledge bases. Such an unweighted directed graph helps to quickly detect multi-word concepts, without performing an exhaustive search throughout all the possible word combinations that can form a commonsense concept.

Single-word concepts, e.g., *house*, that already appear in the clause as a multi-word concept, e.g., *beautiful house*, in fact, are pleonastic (providing redundant information) and are discarded. In this way, the algorithm 2 is able to extract event concepts such as *go market*, *buy some fruits*, *buy fruits*, and *buy vegetables*, representing the concepts to be fed to a common-sense reasoning algorithm for further processing.

```
Data: NounPhrase
Result: Valid object concepts
Split the NounPhrase into bigrams ;
Initialize concepts to Null ;
for each NounPhrase do
    while For every bigram in the NounPhrase do
        POS Tag the Bigram ;
        if adj noun then
            add to Concepts: noun, adj+noun
        else if noun noun then
            add to Concepts: noun+noun
        else if stopword noun then
            add to Concepts: noun
        else if adj stopword then
            continue
        else if stopword adj then
            continue
        else
            Add to Concepts : entire bigram
        end
        repeat until no more bigrams left;
    end
end
```

Algorithm 1: POS-based bigram algorithm

3 Algorithm

First, we extract dependency relations between the words of a sentence. Then, those relations are used to formulate complex concepts. Once, these concepts are extracted we obtain related common-sense knowledge of the concepts from ConceptNet. Below, we first describe the use of the dependency relations to form concepts and latter we discuss how related common-sense knowledge can be inferred from ConceptNet.

3.1 Formation of Concepts using Dependency Relations

Subject noun Rule

Trigger: when the active token is found to be the syntactic subject of a verb.
Behavior: if a word h is in a subject noun relationship with a word t then the concept t-h is extracted.
Example: In (1), *movie* is in a subject relation with *boring*.

(1) The movie is boring.

Here the concept `(boring-movie)` is extracted.

Data: Natural language sentence
Result: List of concepts
Find the number of verbs in the sentence;
for *every clause* **do**
    extract VerbPhrases and NounPhrases;
    stem VERB ;
    for *every NounPhrase with the associated verb* **do**
        find possible forms of *objects* ;
        link all *objects* to stemmed verb to get *events*;
    end
    repeat until no more clauses are left;
end

Algorithm 2: Event concept extraction algorithm

Joint Subject noun and Adjective complement rule

Trigger: when the active token is found to be the syntactic subject of a verb and the verb is on adjective complement relation with an adverb.

Behavior: if a word *h* is in a subject noun relationship with a word *t* and the word *t* is with adjective complement relationship with a word *w* then the concept *w-h* is extracted.

Example: In (2), *flower* is in a subject relation with *smells* and *smells* is in adjective complement relationship with *bad*.

(2) The flower smells bad.

Here the concept `(bad-flower)` is extracted.

Direct nominal objects This complex rule deals with direct nominal objects of a verb.

Trigger: when the active token is head verb of a direct object dependency relation.

Behavior: if a word *h* is in a direct nominal object relationship with a word *t* then the concept *h-t* is extracted.

Example: In (3) the system extracts the concept `(see,movie)`.

(3) Paul saw the movie in 3D.

`(see,in,3D)` is not treated at this stage since it will later be treated by the standard rule for prepositional attachment.

Adjective and clausal complements Rules 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).

Trigger: when the active token is head verb of one of the complement relations.

Behavior: if a word *h* is in a direct nominal object relationship with a word *t* then the concept *h-t* is extracted.

Example: in (4), *smells* is the head of a clausal complement dependency relation with *bad* as the dependent.

(4) This meal smells bad.

In this example the concept `(smell,bad)` is extracted.

Negation Negation is also a crucial components of natural language text which usually flips the meaning of the text. This rule is used to identify whether a word is negated in the text.

Trigger: when in a text a word is negated.

Behavior: if a word *h* is negation by a *negation marker t* then the concept *t-h* is extracted.

Example: in (5), *like* is the head of the negation dependency relation with *not* as the dependent. Here, *like* is negated by the negation marker *not*.

(5) I do not like the movie.

Based on the rule described above the concept `(not, like)` is extracted.

Open clausal complements Open clausal complements are clausal complements of a verb that do not have their own subject, meaning that they (usually) share their subjects with that of the matrix clause. The corresponding rule is complex in the same way as the one for direct objects.

Trigger: when the active token is the head of the relation

Behavior: as for the case of direct objects, the algorithm tries to determine the structure of the dependent of the head verb. Here the dependent is itself a verb, therefore, the system tries to establish whether the dependent verb has a direct object or a clausal complement of its own. In a nutshell, the system is dealing with three elements: the head verb(h), the dependent verb(d), and the (optional) complement of the dependent verb (t). Once these elements have all been identified, the concept (h,d,t) is extracted

Example: in (6), *like* is the head of the *open clausal complements* dependency relation with *praise* as the dependent and the complement of the dependent verb *praise* is *movie*.

(6) Paul likes to praise good movies.

So, in this example the concept `(like,praise,movie)` is extracted.

Modifiers

Adjectival, adverbial and participial modification The rules for items modified by adjectives, adverbs or participles all share the same format.

Trigger: these rules are activated when the active token is modified by an adjective, an adverb or a participle.
Behavior: if a word *w* is modified by a word *t* then the concept *(t,w)* is extracted.
Example: in (7) the concept *bad, loser* is extracted.

(7) a. Paul is a bad loser.

Prepositional phrases Although prepositional phrases do not always act as modifiers we introduce them in this section as the distinction does not really matter for their treatment.

Trigger: the rule is activated when the active token is recognized as typing a prepositional dependency relation. In this case, the head of the relation is the element to which the PP attaches, and the dependent is the head of the phrase embedded in the PP.
Behavior: instead of looking for the complex concept formed by the head and dependent of the relation, the system uses the preposition to build a ternary concept.
Example: in (8), the parser yields a dependency relation typed `prep_with` between the verb *hit* and the noun *hammer* (=the head of the phrase embedded in the PP).

(8) Bob hit Marie with a hammer.

Therefore the system extracts the complex concept `(hit, with, hammer)`.

Adverbial clause modifier This kind of dependency concerns full clauses that act as modifiers of a verb. Standard examples involve temporal clauses and conditional structures.

Trigger: the rule is activated when the active token is a verb modified by an adverbial clause. The dependent is the head of the modifying clause.
Behavior: if a word *t* is a adverbial clause modifier of a word *w* then the concept *(t-w)* is extracted.
Example: in (9), the complex concept `(play,slow)` is extracted.

(9) The machine slows down when the best games are playing.

Noun Compound Modifier

Trigger: the rule is activated when it finds a noun composed with several nouns. A noun compound modifier of an NP is any noun that serves to modify the head noun.
Behavior: if a noun-word *w* is modified by another noun-word *t* then the complex concept *(t-h)* is extracted.
Example: in (10), the complex concept `(birthday,party)` is extracted.

(10) Erik threw the birthday party for his girlfriend.

Single Word Concepts Words having part-of-speech VERB, NOUN, ADJECTIVE and ADVERB are also extracted from the text. Single word concepts which exist in the multi-word-concepts are discarded as they carry redundant information. For example, concept *party* that already appears in the concept *birthday party* so, we discard the concept *party*.

3.2 Obtaining Common-Sense Knowledge from ConceptNet

ConceptNet [13] represents the information from the Open Mind corpus as a directed graph, in which the nodes are concepts and the labeled edges are common-sense assertions that interconnect them. For example, given the two concepts person and cook, an assertion between them is CapableOf, i.e. a person is capable of cooking [13].

After obtaining concepts from the text we send them as queries to ConceptNet. From ConceptNet we find the common-sense-knowledge related to the query concepts. For example, when we send the concept *birthday party* as a query to ConceptNet we get related concepts such as *cake*, *buy present*. From ConceptNet we find the following relations

- cake – AtLocation $\rightsquigarrow$ birthday party.
- buy present – UsedFor $\rightsquigarrow$ birthday party.

These common-sense concepts are used to gather more knowledge about the concepts as they have direct connections with *birthday party*. From ConceptNet we get *cake* is used in *birthday party* and people *buy present* for the *birthday party*. So, this process help us to acquire more knowledge about the concepts we extract by the methodology described in Section 4.1. Hence, the joint exploitation of the extracted concepts and ConceptNet offer machine a better understanding of the natural language text. Our approach enables computer to understand the topic of the text as well as the meaning conveyed by the text.

4 Novelty of Our Work

Existing approaches mainly discuss on the automatic extraction of concepts based on the hyponomy and hypernomy relationship of words in a text. The concepts extracted by their methods can easily identify on which topic the text is all about but cant describe the meaning inferred by the text i.e. using those methods we are unable to know what the text tells about the topic. Such information are often found to be crucial for several cognitive tasks such as sentiment analysis, emotion analysis, opinion mining etc where both topic and meaning of the text are important. Our method is able to extract concepts which carry the meaning expressed by the text as well as our method also extracts the concepts which tells about the topic or theme of the text. The difference between our approach and state of the art can be explained using a simple example (11-a). For (11-a) existing approaches can only extract concepts related to Coffee and Starbucks based on the ontologies the methods use. However, our approach extracts the concepts: *like-coffee*, *coffee-of-Starbucks*, *coffee*, *Starbucks* as well as concepts related to *like-coffee*, *coffee*, *coffee-of-Starbucks* and *Starbucks*. Concepts related to *like-coffee*, *coffee*, *coffee-of-Starbucks* and *Starbucks* are extracted from the ConceptNet ontology. Clearly, the concepts extracted by our approach carry the meaning (here the sentiment of the speaker) expressed by (11-a), while the state of the art approaches fail to do it.

(11) a. I like the coffee of Starbucks.

Readers may be confused our approach with the syntactic ngrams proposed by [32]. Here, we first describe syntactic n-grams and then show the differences between our concept parser and syntactic n-gram. By dependency syntactic n-gram (sn-gram) we understand a subtree of the dependency tree of a sentence that contains n nodes [30]. Sn-grams can be used as features to represent sentences in the same scenarios as conventional n-grams [31]; more specifically, sn-grams represent dependency trees as vectors in the same way as conventional n-grams represent strings of words. However, unlike conventional n-grams [6], sn-grams represent linguistic entities and are thus much more informative and less noisy. While sn-grams go a long way towards linguistically meaningful representation, numerous phenomena from the presence of functional words to synonymous expressions to insignificant details still introduce noise in this representation and prevent semantically similar constructions to be mapped to identical feature vectors. In this work we present near-paraphrasic rules that simplify and normalize the dependency trees in order to reduce synonymous variation and remove insignificant details and thus improve similarity between feature vectors of semantically similar expressions and reduce data sparseness. Another difference is that syntactic n-grams convey all characteristics of basic n-gram whereas our concept parser extracts semantic from the text. Lets discuss the differences between syntactic n-gram and our proposed concept parser through an example [32].

(12) a. I can even now remember the hour from which I dedicated myself to this great enterprise.

Here, extracted syantactic n-grams are [*remember now, now even, remember hour, remember dedicated, dedicated enterprise, enterprise great, remember now even, remember hour dedicated, hour dedicated enterprise, dedicated enterprise great*].

Whereas, extracted concepts by our concept parser are [*even now, even now remember, remember hour, hour, remember from dedicate, dedicate which to enterprise, dedicate myself to enterprise, dedicate to enterprise, great enterprise*].

After sending these concepts as query to conceptnet in order to acquire more commonsense knowledge we obtain the concept list [*even now, even now remember, remember hour, hour, remember from dedicate, dedicate which to enterprise, dedicate myself to enterprise, dedicate to enterprise, great enterprise, still, sixty minute*]. Here, from conceptnet we find commonsense knowledge *still*, *sixty minute* related to the concepts *even now* and *hour* respectively.

Clearly from above examples we see the proposed concept parser is able to extract more semantic. *even now*, *even now remember* extracted by proposed concept parser express more semantic compare to *now even* and *remember now even* extracted by syntactic n-grams.

In (13). our concept parser extracts *food, food smell, bad food, smell bad*. But, syntactic n-gram method extracts *smell bad food, smell bad*. From this example, our concept parser is able to extract good semantic conveyed by *bad food*.

(13) a. The food smells bad.

5 Experiments and Results

To calculate the performance, we selected 300 sentences from the *Stanford Sentiment Dataset* [33] and extracted the concepts manually. This process yielded 3204 concepts. Below in Table 1 we show the accuracy of concept mining process using approach with the POS based approach described in Section 2. concepts in them manually.

Table 1. Results obtained using different algorithms on the dataset

Algorithm	Precision
Part-of-Speech Approach	86.10%
Proposed Approach	92.21%

6 Applications of the Proposed Concept Parser

We used the proposed concept parser in many applications and found it to perform superior to the existing concept parsers. As, to the best of our knowledge Part Of Speech based concept parser has the highest accuracy till now in extracting concepts from text so we compare the result obtained using our concept parser with the Part Of Speech based concept parser. This section also shows the proposed concept parser outperforms Syntactic N-grams [32] technique in these tasks. Syntactic N-grams method uses dependency tree of a text and by following the paths in the tree it extracts ngrams. It is called syntactic because it carries syntactic information of words i.e. information on word relations in a text. But, the method consists all characteristics of the ngrams.

We treated each application as classification task. As discussions on feature extraction process and classification method are out of the scope of this paper, we do not present those details in this paper. Please find those details in [23][22][27].

6.1 Sentiment Analysis of Text

For experiments on detecting positive and negative sentiment in texts, we used Stanford Twitter dataset[11]. We cast this task as a classification task. For sentiment analysis experiment,this was binary classification.We report the results obtained with the Extreme Learning Machine (ELM)[16] as the classifier. Concept parser was used to extract concepts from a text and those concepts were used to form feature vector. Details of the feature formulation is skipped in this paper as this is not the focus of the paper. Table 2 shows the experimental results and comparison between the performance of proposed concept parser and POS based concept parser and Syntactic N-grams in the task.

Table 2. Sentiment analysis on Stanford Twitter dataset

Algorithm	Precision
Syntactic N-grams	83.23%
Part-of-Speech Approach	82.20%
Proposed Approach	85.05%

6.2 Emotion Recognition from Text

As a dataset for the emotion detection experiment, we used the ISEAR dataset. We cast the task as a six-way classification, where the six classes were anger, sadness, disgust, fear, surprise, and joy. This experiment was also based on the concept extraction process from text and the extracted concepts were used to form feature vector to learn the Emotion Recognition classifier. ELM was used as a classifier for this task. Table 3 shows the significant improvement in the accuracy of the Emotion Recognition task when proposed concept parser is used instead of POS based concept parser and syntatic N-grams are used for the task.

Table 3. Emotion detection on the ISEAR dataset

Algorithm	Precision
Syntatic N-grams	61.25%
Part-of-Speech Approach	62.10%
Proposed Approach	63.25%

6.3 Personality Recognition from Text

For experiments on detection personality from text,we used five-way classification according to the five personality traits described by Mathews et al. (2009), which are- openness, conscientiousness, extraversion,agreeableness, and neuroticism, sometimes abbreviated as OCEAN by their first letters. To experiment, we used the dataset provided by [19]. We treated this task as a classification. For this task also, we used concept parser to extract concepts from the text and later they were used to form the features to train the classifier. As a classifier, we used ELM. Table 4 shows the experimental results.

7 Conclusion and Future Work

In this work, we use the dependency relation between words to extract concepts from text. The joint exploitation of these concepts and conceptnet help to acquire more knowledge thus it enable a better understanding of the text. Experiment shows how well

Table 4. Personality detection on the essays dataset for personality detection

	Extraversion	Neuroticism	Agreeableness	Conscientiousness	Openness
Syntatic N-grams	0.532	0.561	0.502	0.566	0.592
Part-of-Speech Approach	0.546	0.557	0.540	0.564	0.604
proposed method	0.634	0.637	0.615	0.633	0.661

it performs and it outperforms state of the art model. Future work involves to discover more useful dependency relationship to mine the concepts. Also, removing the concepts which do not carry good semantic rather carry noise is a challenging task. Along with using conceptnet, how other ontologies can help to enrich the concept mining process is also a big task to deal with. We also aim to use extracted concepts for cognitive tasks such as opinion mining, sentiment analysis, personality detection etc.

Acknowledgements This work was partially supported by the Governments of Mexico and India (grant CONACYT 122030) and Government of Mexico (grant IPN SIP 20144534; COFAA-IPN).

References

1. Eneko Agirre, Olatz Ansa, Eduard Hovy, and David Martínez. Enriching very large ontologies using the www. *arXiv preprint cs/0010026*, 2000.
2. Erik Cambria, Paolo Gastaldo, Federica Bisio, and Rodolfo Zunino. An ELM-based model for affective analogical reasoning. *Neurocomputing*, 2014.
3. Erik Cambria, Amir Hussain, Catherine Havasi, Chris Eckl, and James Munro. Towards crowd validation of the uk national health service. In *WebSci*, Raleigh, 2010.
4. Erik Cambria and Bebo White. Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2), 2014.
5. Cungen Cao, Qiangze Feng, Ying Gao, Fang Gu, Jinxin Si, Yuefei Sui, Wen Tian, Haitao Wang, Lili Wang, Qingtian Zeng, et al. Progress in the development of national knowledge infrastructure. *Journal of Computer Science and Technology*, 17(5):523–534, 2002.
6. Arda Çelebi and Arzucan Özgür. N-gram parsing for jointly training a discriminative constituency parser. *Polibits*, 48:5–12, 2013.
7. WL Chen, JB Zhu, TS Yao, and YX Zhang. Automatic learning field words by bootstrapping. In *Proc. of the JSCL. Beijing: Tsinghua University Press*, volume 72, 2003.
8. Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.
9. Bo Du, HF Tian, Li Wang, and RZ Lu. Design of domain-specific term extractor based on multi-strategy. *Computer Engineering*, 31(14):159–160, 2005.
10. Boris Gelfand, Marilyn Wulfekuler, and WF Punch. Automated concept extraction from plain text. In *AAAI 1998 Workshop on Text Categorization*, pages 13–17, 1998.
11. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.
12. Yannis Harlambous and Vitaly Klyuev. Thematically reinforced explicit semantic analysis. *International Journal of Computational Linguistics and Applications*, 4(1):79–94, 2013.

13. Catherine Havasi, Robert Speer, and Jason Alonso. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, pages 27–29, 2007.
14. Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics, Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.
15. Newton Howard and Erik Cambria. Intention awareness: Improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences*, 3(9), 2013.
16. Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.
17. L Liu, C Cao, and H Wang. Acquiring hyponymy relations from large chinese corpus. *WSEAS Transactions on Business and Economics*, 2(4):211, 2005.
18. Lei Liu, Cun-gen Cao, Hai-tao Wang, and Wei Chen. A method of hyponym acquisition based on "isa" pattern. *Journal of Computer Science*, pages 146–151, 2006.
19. François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *J. Artif. Intell. Res.(JAIR)*, 30:457–500, 2007.
20. Keiichi Nakata, Angi Voss, Marcus Juhnke, and Thomas Kreifelts. Collaborative concept extraction from documents. In *In Proceedings of the 2nd Int. Conf. on Practical Aspects of Knowledge management (PAKM 98*. Citeseer, 1998.
21. Chris D. Paice. Another stemmer. *ACM SIGIR Forum*, 24(3):56–61, 1990.
22. Soujanya Poria, Erik Cambria, Amir Hussain, and Guang-Huang Bin. Big multimodal data analysis. *Elsevier Neural Networks*, 2014.
23. Soujanya Poria, Alexandar Gelbukh, Basant Agarwal, Erik Cambria, and Newton Howard. Common sense knowledge based personality recognition from text. In *Advances in Soft Computing and Its Applications*, pages 484–496. Springer, 2013.
24. Soujanya Poria, Alexander Gelbukh, Erik Cambria, Dipankar Das, and Sivaji Bandyopadhyay. Enriching senticnet polarity scores through semi-supervised fuzzy clustering. In *2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW)*, pages 709–716. IEEE, 2012.
25. Soujanya Poria, Alexander Gelbukh, Erik Cambria, Peipei Yang, Amir Hussain, and Tariq Durrani. Merging senticnet and wordnet-affect emotion lists for sentiment analysis. In *2012 IEEE 11th International Conference on Signal Processing (ICSP)*, volume 2, pages 1251–1255. IEEE, 2012.
26. Soujanya. Poria, Alexander. Gelbukh, Amir. Hussain, Newton. Howard, Dipankar. Das, and Sivaji. Bandyopadhyay. Enhanced senticnet with affective labels for concept-based opinion mining. *Intelligent Systems, IEEE*, 28(2):31–38, 2013.
27. Soujanya Poria, Gregoire Winterstein, Erik Cambria, Alexander Gelbukh, Amir Hussain, and Guang-Huang Bin. Dependency-based rules for concept-level sentiment analysis. *Elsevier Knowledge-Based Systems*, 2014.
28. Dheeraj Rajagopal, Erik Cambria, Daniel Olsher, and Kenneth Kwok. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 565–570. International World Wide Web Conferences Steering Committee, 2013.
29. Paul M Ramirez and Chris A Mattmann. Ace: improving search engines via automatic concept extraction. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004 (IRI 2004)*, pages 229–234. IEEE, 2004.
30. Grigori Sidorov. Non-continuous syntactic n-grams. *Polibits*, 48:67–75, 2013.

31. Grigori Sidorov. Syntactic dependency based n-grams in rule based automatic English as second language grammar correction. *International Journal of Computational Linguistics and Applications*, 4(2):169–188, 2013.
32. Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860, 2014.
33. Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. EMNLP, 2013.
34. Paola Velardi, Paolo Fabriani, and Michele Missikoff. Using text processing techniques to automatically enrich a domain ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems, Volume 2001*, pages 270–284. ACM, 2001.
35. Zhang Yuntao, Gong Ling, Wang Yongcheng, and Yin Zhonghang. An effective concept extraction method for improving text classification performance. *Geo-Spatial Information Science*, 6(4):66–72, 2003.
36. JH Zheng and JL Lu. Study of an improved keywords distillation method. *Computer Engineering*, 31(18):194–196, 2005.