A graph-based approach to commonsense concept extraction and semantic similarity detection

Conference Paper · May 2013

DOI: 10.1145/2487788.2487995

4 authors, including:

Dheeraj Rajagopal
Agency for Science, Technology and Research (A*STAR)
30 PUBLICATIONS **931** CITATIONS

SEE PROFILE

Erik Cambria
Nanyang Technological University
574 PUBLICATIONS **47,549** CITATIONS

SEE PROFILE

Kenneth Kwok
Institute Of High Performance Computing
37 PUBLICATIONS **1,749** CITATIONS

SEE PROFILE

A Graph-Based Approach to Commonsense Concept Extraction and Semantic Similarity Detection

Dheeraj Rajagopal
Temasek Laboratories
National University
of Singapore
dheeraj@nus.edu.sg

Erik Cambria
Temasek Laboratories
National University
of Singapore
cambria@nus.edu.sg

Daniel Olsher
Temasek Laboratories
National University
of Singapore
olsher@nus.edu.sg

Kenneth Kwok
Temasek Laboratories
National University
of Singapore
kenkwok@nus.edu.sg

ABSTRACT

Commonsense knowledge representation and reasoning support a wide variety of potential applications in fields such as document auto-categorization, Web search enhancement, topic gisting, social process modeling, and concept-level opinion and sentiment analysis. Solutions to these problems, however, demand robust knowledge bases capable of supporting flexible, nuanced reasoning. Populating such knowledge bases is highly time-consuming, making it necessary to develop techniques for deconstructing natural language texts into commonsense concepts. In this work, we propose an approach for effective multi-word commonsense expression extraction from unrestricted English text, in addition to a semantic similarity detection technique allowing additional matches to be found for specific concepts not already present in knowledge bases.

Categories and Subject Descriptors

H.3.1 [**Information Systems Applications**]: Linguistic Processing; I.2.7 [**Natural Language Processing**]: Language parsing and understanding

General Terms

Algorithms

Keywords

AI; Commonsense knowledge representation and reasoning; Natural language processing; Semantic similarity

1. INTRODUCTION

Commonsense knowledge describes basic knowledge and understandings that people acquire through experience, e.g., "something sharp might cut your skin, if it is not handled carefully", "people don't like to be repeatedly interrupted", "it's better not to touch a hot stove", or "if you cross the road when the signal is still red, you are breaking the law".

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

Commonsense reasoning problems are often solved by populating knowledge bases with commonsense information and then executing reasoning algorithms drawing on this knowledge in order to formulate new conclusions. Such information may be represented via the use of traditional predicate logic statements [15, 11] or by the use of natural-language-based semantic networks [23, 3, 19]. A commonsense fact such as "a couch is something for sitting on", for example, is usually represented as `Couch HasProperty Sit`.

It is clear, then, that semantic parsing, i.e., the deconstruction of text into multiple-word concepts, is a key step in applying commonsense reasoning to natural language processing and understanding, as shown by recent approaches to concept-level opinion and sentiment analysis [5, 12, 18]. Parsing, moreover, should be as time- and resource-efficient as possible, enabling tasks such as real-time human-computer interaction (HCI) [2] and big social data analysis [4].

In this work, we propose a graph-based technique for effectively and quickly identifying event and object concepts in open English text. The technique is able to draw upon pre-existing knowledge bases, using syntactic and semantic matching to augment results with related multi-word expressions. The paper is organized as follows: section 2 introduces related work, section 3 and 4 describe in detail how concept extraction and semantic similarity detection are performed, section 5 evaluates the proposed approach, and section 6 offers concluding remarks and discusses avenues for future work.

2. RELATED WORK

Commonsense knowledge parsing can be performed using a combination of syntax and semantics, via syntax alone (making use of phrase structure grammars), or statistically, using classifiers based on training algorithms. Construction-based parsing [17, 4] offers high semantic sensitivity, the ability to extract knowledge from grammatically-incorrect text, and can use world knowledge to choose the most likely parses, but requires access to construction corpora.

The Open Mind Common Sense (OMCS) project [23] uses a syntactical parsing technique that compares natural language sentences against regular expression patterns for collecting specific pieces of commonsense knowledge.

OMCS employs a purely syntactical approach encompassing stopwords, punctuation removal, word stemming to identify commonsense concepts. Part-of-speech (POS) tagging involves annotating syntactic structure with language-specific parts of speech. Related work includes tag sequence probability [7], while more recent approaches use lexical probabilities [26]. Statistical parsing has been possibly the most widely adopted technique for collecting information from text [8], together with active learning, which aims to select effective features [13, 25]

With respect to semantic similarity detection, previous work has mainly employed machine learning techniques such as support vector machines [14], latent semantic indexing [9], linear discriminant analysis [5], and kernel functions [22].

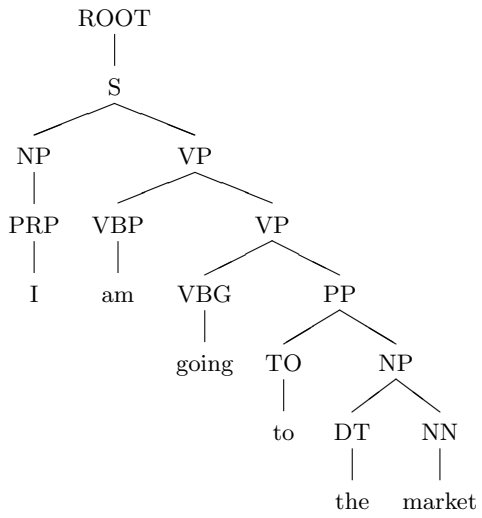
3. CONCEPT EXTRACTION

The aim of the proposed concept extraction technique is to break text into clauses and, hence, deconstruct such clauses into small bags of concepts (SBoC) [3], in order to feed these into a commonsense reasoning algorithm. For applications in fields such as real-time HCI 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 (affect) associated to such semantics are often enough to quickly perform tasks such as emotion recognition and polarity detection.

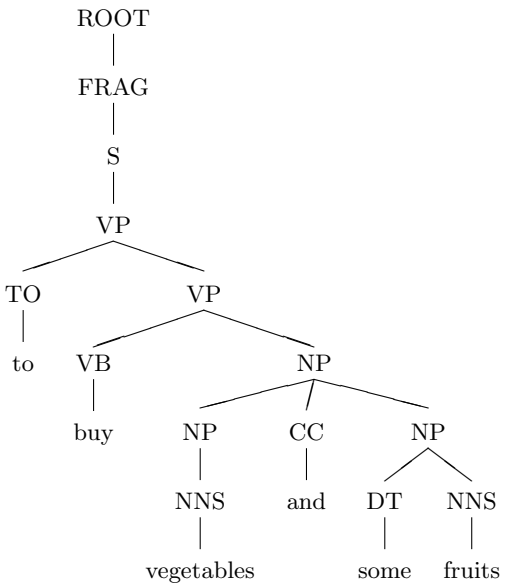
3.1 From Sentence to Verb and Noun Chunks

The first step in the proposed algorithm 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 [16] is used to chunk the input text. A sentence like "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 following parse tree:



and



3.2 Obtaining the Full List of Concepts

Next, clauses are normalized in two stages. First, each *verb* chunk is normalized using the Lancaster stemming algorithm [20]. 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 commonsense 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. ADJECTIVE NOUN : The adj+noun combination and noun as a stand-alone concept are added to the objects list.
2. ADJECTIVE STOPWORD : The entire bigram is discarded.
3. NOUN ADJECTIVE : 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 ADJECTIVE: 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.

```
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

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 (Fig. 1).

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 SBoCs to be fed to a commonsense reasoning algorithm for further processing.

```
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

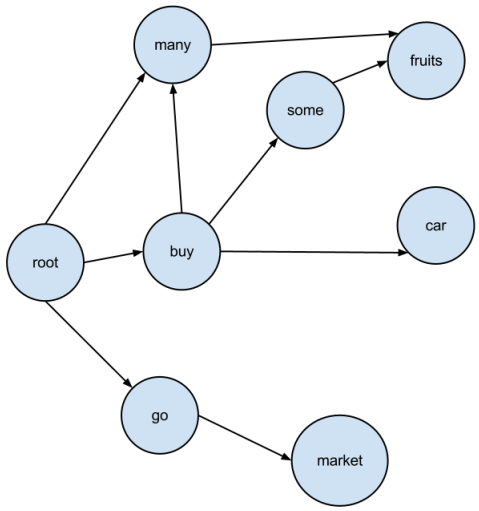


Figure 1: Example parse graph for multi-word expressions

4. SIMILARITY DETECTION

Because natural language concepts may be expressed in a multitude of forms, it is necessary to have a technique for defining the similarity of multi-word expressions so that a concept can be detected in all its different forms.

The main aim of the proposed similarity detection technique, in fact, is to find concepts that are both syntactically and semantically related to the ones generated by the event concept extraction algorithm, in order to make up for concepts for which no matches are found in the knowledge bases. In particular, the POS tagging based bigram algorithm is employed to calculate syntactic matches, while the knowledge bases are exploited to find semantic matches.

Beyond this, concept similarity may be exploited to merge concepts in the database and thus reduce data sparsity. When commonsense data is collected from different data sources, in fact, the same concepts tend to appear in different forms and merging these can be key for enhancing the commonsense reasoning capabilities of the system.

4.1 Syntactic Match Step

The syntactic match step checks whether two concepts have at least one object in common. For each noun phrase, objects and their matches from the knowledge bases are extracted, providing a collection of related properties for specific concepts. All the matching properties for each noun phrase are collected separately. The sets are then compared in order to identify common elements. If common elements exist, phrases are considered to be similar. We deduce such similarity as shown in Algorithm 3.

4.2 Semantic Similarity Detection

Semantic similarity can be calculated by means of multi-dimensional scaling [5] on a matrix whose rows are natural language concepts (e.g., *dog* or *bake cake*), whose columns are commonsense features (e.g., *isA-pet* or *hasEmotion-joy*), and whose entries indicate truth values of assertions.

Data: NounPhrase1, NounPhrase2
Result: *True* if the concepts are similar, else *False*
if *Both phrases have atleast one* noun *in common* **then**
  Objects1 := All Valid Objects for NounPhrase1;
  Objects2 := All Valid Objects for NounPhrase2;
  M1 = matches from KB for
  M1 := ∅ ;
  M2 := ∅ ;
  for *all* concepts *in NounPhrase1* **do**
    M1 := M1 ∪ *all property matches for concept*;
  end
  for *all* concepts *in NounPhrase2* **do**
    M2 := M2 ∪ *all property matches for concept* ;
  end
  SetCommon = M1 ∪ M2 ;
  if *length of* SetCommon > *0* **then**
    The Noun Phrases are similar
  else
    They are not similar
  end

Algorithm 3: Finding similar concepts

In particular, we use AffectNet[1] [3] (hereafter termed A), a 14,301 × 117,365 matrix of affective commonsense knowledge. In A, each concept is represented by a vector in the space of possible features whose values are positive for features that produce an assertion of positive valence (e.g., "a penguin is a bird"), negative for features that produce an assertion of negative valence (e.g., "a penguin cannot fly"), and zero when nothing is known about the assertion.

The degree of similarity between two concepts, then, is the dot product between their rows in A. The value of such a dot product increases whenever two concepts are described by the same features and decreases when they are described by features that are negations of each other. In particular, we use truncated singular value decomposition (SVD) [27]: the resulting matrix[2] has the form $\tilde{A} = U_k \, \Sigma_k \, V_k^T$ and is a low-rank approximation of A, the original data. This approximation is based on minimizing the Frobenius norm of the difference between A and $\tilde{A}$ under the constraint $rank(\tilde{A}) = k$. For the Eckart-Young theorem [10], it represents the best approximation of A in the least-square sense, in fact:

$$\begin{aligned} \min_{\tilde{A}|rank(\tilde{A})=k} |A - \tilde{A}| &= \min_{\tilde{A}|rank(\tilde{A})=k} |\Sigma - U^* \tilde{A} V| \\ &= \min_{\tilde{A}|rank(\tilde{A})=k} |\Sigma - S| \end{aligned}$$

assuming that $\tilde{A}$ has the form $\tilde{A} = USV^*$, where S is diagonal. From the rank constraint, i.e., S has k non-zero diagonal entries, the minimum of the above statement is obtained as follows:

$$\min_{\tilde{A}|rank(\tilde{A})=k} |\Sigma - S| = \min_{s_i} \sqrt{\sum_{i=1}^{n} (\sigma_i - s_i)^2} =$$

$$= \min_{s_i} \sqrt{\sum_{i=1}^{k} (\sigma_i - s_i)^2 + \sum_{i=k+1}^{n} \sigma_i^2} = \sqrt{\sum_{i=k+1}^{n} \sigma_i^2}$$

[1] http://sentic.net/affectnet.zip
[2] http://sentic.net/affectivespace.zip

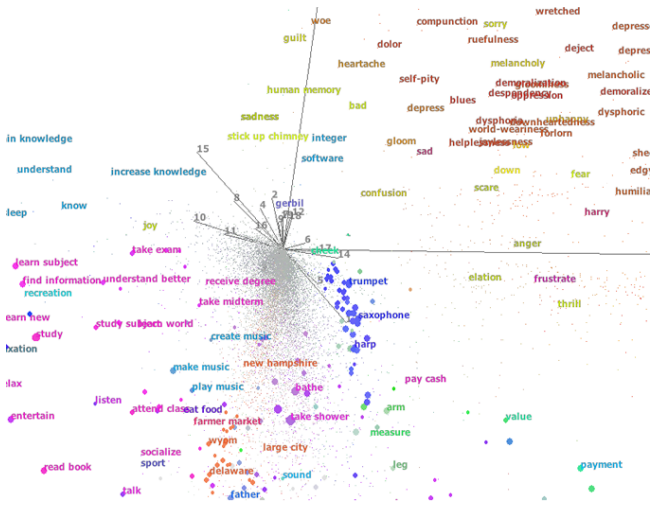


Figure 2: Vector space model for reasoning on the semantic relatedness of commonsense concepts

Therefore, $\tilde{A}$ of rank k is the best approximation of A in the Frobenius norm sense when $\sigma_i = s_i$ $(i = 1, ..., k)$ and the corresponding singular vectors are the same as those of A. If we choose to discard all but the first k principal components, commonsense concepts are represented by vectors of k coordinates, which can be seen as describing multi-word expressions in terms of 'eigenconcepts' that form the axes of the resulting vector space, i.e., its basis e_0,...,e_{k-1} (Fig. 2).

Thus, by exploiting the information sharing property of truncated SVD, concepts with similar meaning are likely to have similar semantic features - that is, concepts conveying the same semantics tend to fall near each other in the space. Concept similarity does not depend on their absolute positions in the vector space, but rather on the angle they make with the origin. In order to measure such semantic relatedness, then, $\tilde{A}$ is clustered by using a k-medoid approach [21]. Differently from the k-means algorithm (which does not pose constraints on centroids), k-medoids do assume that centroids must coincide with k observed points. The k-medoids approach is similar to the partitioning around medoids (PAM) algorithm, which determines a medoid for each cluster selecting the most centrally located centroid within that cluster.

Unlike other PAM techniques, however, the k-medoids algorithm runs similarly to k-means and, hence, requires a significantly reduced computational time. Given that the distance between two points in the space is defined as $D(e_i, e_j) = \sqrt{\sum_{s=1}^{d'} \left(e_i^{(s)} - e_j^{(s)}\right)^2}$, the adopted algorithm can be summarised as follows:

1. Each centroid $\bar{e}_i \in \mathbb{R}^{d'}$ $(i = 1, 2, ..., k)$ is set as one of the k most representative instances of general categories such as time, location, object, animal, and plant;
2. Assign each instance e_j to a cluster $\bar{e}_i$ if $D(e_j, \bar{e}_i) \leq D(e_j, \bar{e}_{i'})$ where $i(i') = 1, 2, ..., k$;
3. Find a new centroid $\bar{e}_i$ for each cluster c so that $\sum_{j \in Cluster\ c} D(e_j, \bar{e}_i) \leq \sum_{j \in Cluster\ c} D(e_j, \bar{e}_{i'})$;
4. Repeat step 2 and 3 until no changes are observed.

5. EVALUATION

A manually labeled dataset of 200 multi-word concept pairs were considered for evaluating the similarity detection capabilities of the developed parser[3].

Such pairs were randomly selected from a collection of different knowledge bases, namely: DBPedia [1], ConceptNet [23], NELL [6], and tuples obtained from Google N-grams [24], containing about 9 million triples. As a baseline, we considered the syntactic similarity algorithm alone; then, we calculated semantic similarity by means of multi-dimensional scaling; finally, an ensemble of both approaches was employed. Results are shown in Table 1.

A goldset of 50 natural language sentences, moreover, was selected to test how the ensemble application of concept extraction and semantic similarity detection can improve semantic parsing, with respect to the POS-based bigram algorithm alone and a naïve parser that only uses the Stanford Chunker and the Lancaster stemming algorithm. Results are listed in Table 2.

Algorithm	Precision	Recall	F-measure
Syntactic similarity	65.6%	67.3%	66.4%
Semantic similarity	77.2%	70.8%	73.9%
Ensemble similarity	85.4%	74.0%	79.3%

Table 1: Performance of different similarity detection algorithms over 200 concept pairs

Algorithm	Concept extraction accuracy
Naïve parser	65.8%
POS-based bigram	79.1%
POS-based + similarity	87.6%

Table 2: Performance of different parsing algorithms over 50 natural language sentences

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel approach for effectively extracting event and object concepts from natural language text, aided by a semantic similarity detection technique capable of effectively finding syntactically and semantically related concepts. We also explored how knowledge may be used to expand the reach of matching algorithms and compensate for database sparsity.

Future work will involve exploration of how commonsense knowledge may be repurposed to generate even more knowledge by using existing commonsense to detect natural language patterns and, hence, match such patterns on new texts in order to extract previously unknown pieces of knowledge.

In addition, work will be undertaken exploring how to create ad hoc knowledge extraction algorithms that output data ideal for immediate entry into specific commonsense knowledge representation and reasoning systems.

7. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735, 2007.

[3] `http://sentic.net/parser.zip`

[2] E. Cambria, N. Howard, J. Hsu, and A. Hussain. Sentic blending: Scalable multimodal fusion for continuous interpretation of semantics and sentics. In *IEEE SSCI*, Singapore, 2013.

[3] E. Cambria and A. Hussain. *Sentic Computing: Techniques, Tools, and Applications*. Springer, Dordrecht, Netherlands, 2012.

[4] E. Cambria, D. Rajagopal, D. Olsher, and D. Das. Big social data analysis. In R. Akerkar, editor, *Big Data Computing*, chapter 13. Chapman and Hall/CRC, 2013.

[5] E. Cambria, Y. Song, H. Wang, and N. Howard. Semantic multi-dimensional scaling for open-domain sentiment analysis. *IEEE Intelligent Systems*, doi: 10.1109/MIS.2012.118, 2013.

[6] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka, and T. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, pages 1306–1313, Atlanta, 2010.

[7] G. Carroll and E. Charniak. Two experiments on learning probabilistic dependency grammars from corpora. AAAI technical report WS-92-01, Department of Computer Science, Univ., 1992.

[8] E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *AAAI*, pages 598–603, Providence, 1997.

[9] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[10] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[11] C. Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

[12] M. Grassi, E. Cambria, A. Hussain, and F. Piazza. Sentic web: A new paradigm for managing social media affective information. *Cognitive Computation*, 3(3):480–489, 2011.

[13] R. Hwa. Sample selection for statistical grammar induction. In *EMNLP*, pages 45–52, Hong Kong, 2000.

[14] J. Kandola, J. Shawe-Taylor, and N. Cristianini. Learning semantic similarity. *Advances in neural information processing systems*, 15:657–664, 2002.

[15] D. Lenat and R. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Boston, 1989.

[16] C. Manning. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 6608 of *Lecture Notes in Computer Science*, pages 171–189. Springer, 2011.

[17] D. Olsher. COGPARSE: Brain-inspired knowledge-driven full semantics parsing. In *Advances in Brain Inspired Cognitive Systems*, pages 1–11, 2012.

[18] D. Olsher. Full spectrum opinion mining: Integrating domain, syntactic and lexical knowledge. In *ICDM SENTIRE*, pages 693–700, Brussels, 2012.

[19] D. Olsher. COGVIEW & INTELNET: Nuanced energy-based knowledge representation and integrated cognitive-conceptual framework for realistic culture, values, and concept-affected systems simulation. In *IEEE SSCI*, Singapore, 2013.

[20] C. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.

[21] H. Park and C. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

[22] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *WWW*, pages 377–386, Edinburgh, 2006.

[23] R. Speer and C. Havasi. ConceptNet 5: A large semantic network for relational knowledge. In E. Hovy, M. Johnson, and G. Hirst, editors, *Theory and Applications of Natural Language Processing*, chapter 6. Springer, 2012.

[24] N. Tandon, D. Rajagopal, and G. De Melo. Markov chains for robust graph-based commonsense information extraction. In *COLING*, pages 439–446, Mumbai, 2012.

[25] M. Tang, X. Luo, S. Roukos, et al. Active learning for statistical natural language parsing. In *ACL*, pages 120–127, Philadelphia, 2002.

[26] K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*, pages 173–180, Stroudsburg, 2003.

[27] M. Wall, A. Rechtsteiner, and L. Rocha. Singular value decomposition and principal component analysis. In D. Berrar, W. Dubitzky, and M. Granzow, editors, *A Practical Approach to Microarray Data Analysis*, pages 91–109. Springer, 2003.