Commonsense-Based Topic Modeling

Dheeraj Rajagopal
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
dheeraj@nus.edu.sg

Daniel Olsher
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
olsher@nus.edu.sg

Erik Cambria
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
cambria@nus.edu.sg

Kenneth Kwok
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
kenkwok@nus.edu.sg

ABSTRACT

Topic modeling is a technique used for discovering the abstract 'topics' that occur in a collection of documents, which is useful for tasks such as text auto-categorization and opinion mining. In this paper, a commonsense knowledge based algorithm for document topic modeling is presented. In contrast to probabilistic models, the proposed approach does not involve training of any kind and does not depend on word co-occurrence or particular word distributions, making the algorithm effective on texts of any length and composition. 'Semantic atoms' are used to generate feature vectors for document concepts. These features are then clustered using group average agglomerative clustering, providing much improved performance over existing algorithms.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*

General Terms

Algorithms

Keywords

AI, NLP, KR, Topic Modeling, Commonsense Knowledge

1. INTRODUCTION

Topics, defined as distributions over words [7], facilitate keyword-based text mining, document search, and meaning-based retrieval [4]. Probabilistic topic models, such as latent Dirichlet allocation [7], are used to facilitate document queries [39], document comprehension [27], and tag-based recommendations [22]. However, typical *bag-of-words* probabilistic models have several shortcomings.

WISDOM'13 August 11, 2013, Chicago, IL, USA

Firstly, in such models words in a document are considered to be independent from each other, which is a very uncommon scenario in practice. For example, the multi-word expression *"score home run"*, taken as a unit, is clearly related to baseball, but word-by-word conveys totally different semantics. Similarly, in a bag-of-words model, the phrase *"getting fired"* [25] would not convey the meaning "fired from a job". Secondly, removing stopwords from documents often leads to the loss of key information, e.g., in the case of the concept *"get the better of"*, which would become *"get better"* with all stopwords removed, or the multi-word expression *"let go of the joy"*, which would be reduced to simply *"joy"*, reversing the meaning.

The alternative approach we put forth here draws on knowledge of *word meanings*, encoded in a commonsense knowledge database structured in the INTELNET formalism [32], to provide enhanced topic modeling capabilities. In contrast to probabilistic models, our approach does not involve training of any kind and does not depend on word co-occurrence or particular word distributions, making the algorithm effective on texts of any length and composition.

Commonsense knowledge, defined as the basic understanding of the world humans acquire through day-to-day life, includes information such as *"one becomes elated when one sees one's ideas become reality"* and *"working to be healthy is a positive goal"*. Statistical text models can identify email spam and find syntactic patterns in documents, but fail to understand poetry, simple stories, or even friendly e-mails. Commonsense knowledge is invaluable for nuanced understanding of data, with applications including textual affect sensing [24], handwritten text recognition [38], story telling [23], situation awareness in human-centric environments [20], casual conversation understanding [12], social media management [14], opinion mining [9], and more.

This paper proposes a method for using commonsense knowledge for topic modeling. Instead of the 'bag-of-words' model, we propose the *bag-of-concepts* [8], which considers each lexical item as an index to a set of 'semantic atoms' describing the concept referred to by that lexical item. A 'semantic atom' is a small piece of knowledge about a particular concept, perhaps that it tends to have a particular size or be

associated with other known concepts. Taken together, the set of semantic atoms for a concept provides an excellent picture of the practical meaning of that concept. Access to this knowledge permits the creation of 'smart' topic modeling algorithms that take semantics into account, offering improved performance over probabilistic models.

The paper is organized as follows: Section 2 describes related work in the field of topic modeling; Section 3 illustrates how the bags-of-concepts are extracted from natural language text; Section 4 discusses the proposed topic modeling algorithm; Section 5 presents evaluation results; finally, Section 6 concludes the paper and suggests directions for future work.

2. RELATED WORK

In topic modeling literature, a document is usually represented as a word vector $W = \{w_1, w_2, w_3 ....., w_n\}$ and the corpus as a set of documents $C = \{W_1, W_2, W_3 ....., W_n\}$, while the distribution of a topic z across C is usually referred as θ. Common approaches include mixture of unigrams, latent semantic indexing (LSI), and latent Dirichlet allocation (LDA).

2.1 Mixture of Unigrams

The mixture-of-unigrams model [30] is one of the earliest probabilistic approaches to topic modeling. It assumes that each document covers only a *single* topic (an assumption that does not often hold). The probability of a document in this model, constrained by the aim of generating N words independently, is defined as:

$$p\left(W\right) = \sum_{z=1}^{k} \left[\prod_{n=1}^{N} p\left(w_n|z\right) \right] p\left(z\right) \tag{1}$$

where $p\left(W\right)$ is the probability of the document, $p\left(w_n|z\right)$ is the probability of the word conditional on topic, and $p\left(z\right)$ is the probability of the topic. This model has the serious shortcoming of attempting to fit a single topic for the whole document.

2.2 Latent Semantic Indexing

A subsequent major development in topic modeling has been achieved through LSI, most commonly in the form of Hofmann's pLSI algorithm[19], given as:

$$p\left(W, w_n\right) = p\left(W\right) \sum_{z=1}^{k} p\left(w_n|z\right) p\left(z|W\right) \tag{2}$$

where $p\left(z|d\right)$ is the probability of the topic conditional on document. In pLSI, each document is modeled as a bag-of-words and each word is assumed to belong to a particular topic. The algorithm overcomes the shortcoming of the mixture-of-unigrams model by assuming that a document can cover multiple topics. However, it suffers from issues related to data overfitting.

2.3 Latent Dirichlet Allocation

LDA [7] is the state-of-the-art approach to topic modeling. It is a mixed-membership model, which posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. In particular, for a set of N topics:

$$p\left(w\right) = \int_{\theta} \left(\prod_{n=1}^{N} \sum_{z_n=1}^{k} p\left(w_n|z_n;\beta\right) p\left(z_n|\theta\right) \right) p(\theta;\alpha) d\theta \tag{3}$$

where α is the Dirichlet prior parameter per document topic distribution and β is the Dirichlet prior parameter per topic word distribution. Specifically, the LDA algorithm [5] operates as follows:

1. For each topic,
   (a) Draw a distribution over words $\overrightarrow{\beta_v} \sim Dir_v\left(\eta\right)$
2. For each document,
   (a) Draw a vector of topic proportion $\theta_d \sim Dir\left(\overrightarrow{\alpha}\right)$
   (b) For each word,
      i. Draw a topic assignment $Z_{d,n} \sim Mult\left(\overrightarrow{\theta_d}\right)$, $Z_{d,n} \in \{1, ... K\}$
      ii. Draw a word $W_{d,n} \sim Mult\left(\overrightarrow{\beta_{z_{d,n}}}\right)$, $W_{d,n} \in \{1, ... V\}$

To perform topic modeling for a particular task, posterior inference is perfomed using any one of the following methods: mean field variational [6], expectation propagation [28], collapsed Gibbs sampling [15] , distributed sampling [29], collapsed variational inference [37], online variational inference [18] and/or factorization-based inference [1].

Although LDA overcomes shortcomings such as overfitting and the presence of multiple topics within documents, it still uses the bag-of-words model and, hence, wrongly assumes that every word in the document is independent. For this reason, model performance is heavily dependent on topic diversity and corpus volume. LDA and other probabilistic models, in fact, fail to deal with short texts, unless they are trained extensively with all the documents covering the scope of the test document. The proposed commonsense-based model, instead, does not involve training of any kind and does not depend on word co-occurrence or particular word distributions, making the algorithm effective on texts of any length and composition.

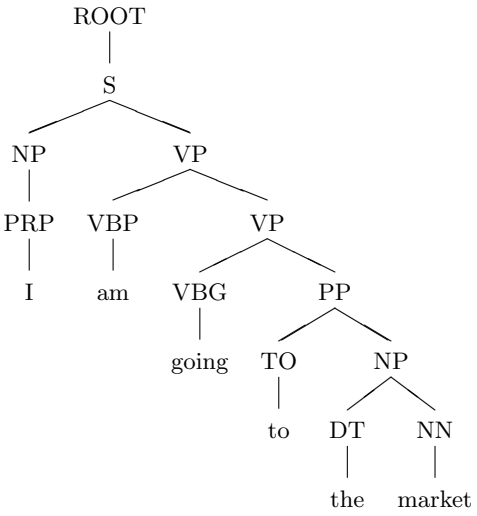
3. SEMANTIC PARSING

The first step to commonsense-based topic modeling is bag-of-concepts extracton. This is performed through a graph-based approach to commonsense concept extraction [34], which breaks sentences into chunks first and then extracts concepts by selecting the best match from a parse graph that maps all the multi-word expressions contained in the commonsense knowledge base.

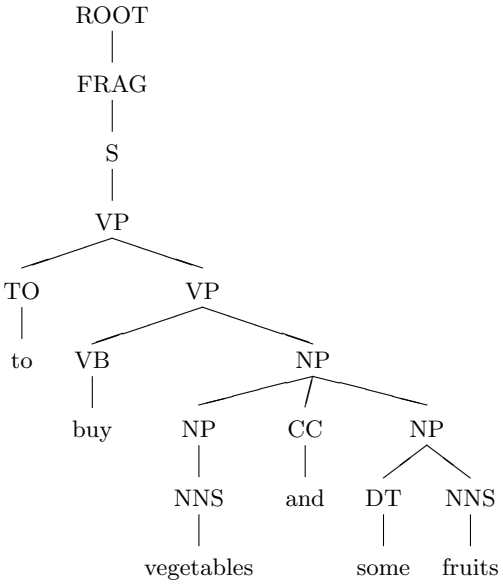
3.1 From Sentence to Verb and Noun Chunks

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 [26] 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 [33]. 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'.

```
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

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: 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

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, 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.

4. TOPIC MODELING ALGORITHM

Once natural language text is deconstructed into bags-of-concepts by means of the graph-based approach to commonsense concept extraction, the topic modeling algorithm determines the final topic set, according to the semantic features associated with the input text.

In the INTELNET architecture [32], knowledge base concepts are defined by the ways in which they *interact* with one another, and semantic features are derived from interconnections as described below. The nuanced structure of the knowledge representation enables us to select just that information most likely to help achieve specific tasks. The rich semantic tapestry generated by the knowledge representation allows us to interpret documents as *collections of inter-connected concepts* rather than independent bags-of-words. The system presently only considers concepts that are in the knowledge base, but as coverage is quite extensive (currently around 9M pieces of information). Knowledge base concepts are collected from sources including ConceptNet [35], DBPedia [2], NELL [10], and WordNet [13].

The presence of a very wide diversity of concepts in the knowledge base makes the model effective for any generic dataset (newspaper articles, reviews, and so on). Documents do not need to be grammatically well-formed, and the algorithm's simultaneous consideration of the semantics of multiple concepts at the same time makes it resistant to

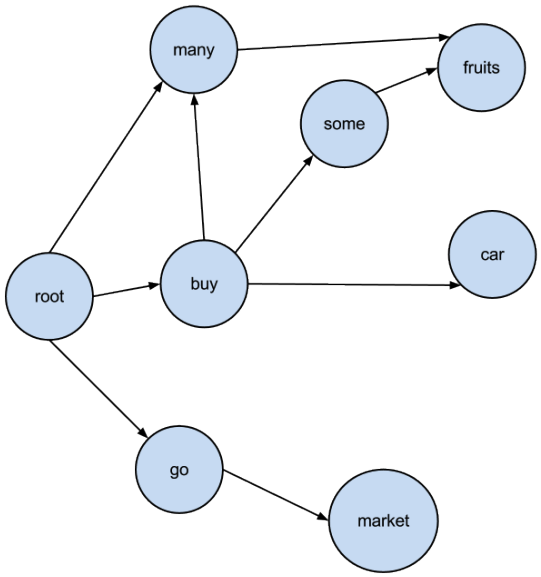


Figure 1: Sample parse graph for multi-word expressions in the knowledge base

spelling errors. Even if one concept is misspelled, it is likely that others from the same topic will be present, thus filling the gap. In domains where documents may be frequently expected to employ concepts not present in general commonsense knowledge bases, a key benefit of INTELNET is its ability to absorb new and noisy knowledge without affecting other knowledge. In these domains, specialized knowledge extracts may be added to the database in order to further enhance performance without damaging general capabilities. These capabilities make feature extraction an ideal base for clustering algorithms.

4.1 The Bag-of-Concepts Model

In our model, every document is represented as a *bag-of-concepts*, where concepts may be single lexical items or multi-word expressions. Phrases like *"get good grade"* are considered to be single concepts, maintaining the semantic interrelatedness of constituent lexical items, unlike the bag-of-words model where every single word is considered to be independent. Documents are defined as the union of the set of commonsense knowledge items retrieved for each individual document concept.

The bag-of-concepts model is different from other language modeling techniques such as *N-grams* wherein the probability of a sequence of words is calculated using the conditional probability of previous words. With bag-of-concepts, the unique semantics attached to particular combinations of words are retained and used to enhance algorithm performance. Specific word sequences evoke unique commonsense concepts together with the particularized semantics attached to those sequences.

The commonsense reasoning framework acts as a hidden layer of knowledge. Concepts in the knowledge base are selected by document contents and are semantically connected to one another, providing a rich data source for clustering.

4.2 Commonsense-Based Feature Extraction

Topics exhibit semantic coherence in that they tend to generate lexical items and phrases with related semantics. Most words related to the same topic tend to share some semantic characteristics. Our commonsense-based approach is similar to the process undertaken by humans when finding similar items - we look at what the *meanings* of the items have in common.

Thus, under our model, *topics* are not discovered merely based on document co-occurrence, but rather by considering the definitive semantic character of constituent concepts.

In INTELNET knowledge bases, concepts inter-define one another, with directed edges indicating semantic dependencies between concepts. In the present algorithm, the features for any particular concept C are defined as the set of concepts reachable via outbound edges from C. Put differently, for each input concept we retrieve those other concepts which, collectively, generate the core semantics of the input concept.

4.3 Clustering and Topic Detection

Our algorithm uses clustering to generate topics from semantic features. Based on experiments with various clustering algorithms, e.g., k-means [17] and expectation-maximization (EM) clustering [11], we determined that group average agglomerative clustering (GAAC) provides the highest accuracy.

GAAC partitions data into trees [3] containing *child* and *sibling* clusters. It generates dendrograms specifying nested groupings of data at various levels [21]. During clustering, documents are represented as vectors of commonsense concepts. For each concept, the corresponding features are extracted from the knowledge base. The proximity matrix is constructed with concepts as rows and features as columns. If a feature is an outbound link of a concept, the corresponding entry in the matrix is 1, otherwise it is 0. Cosine distance is used as the distance metric.

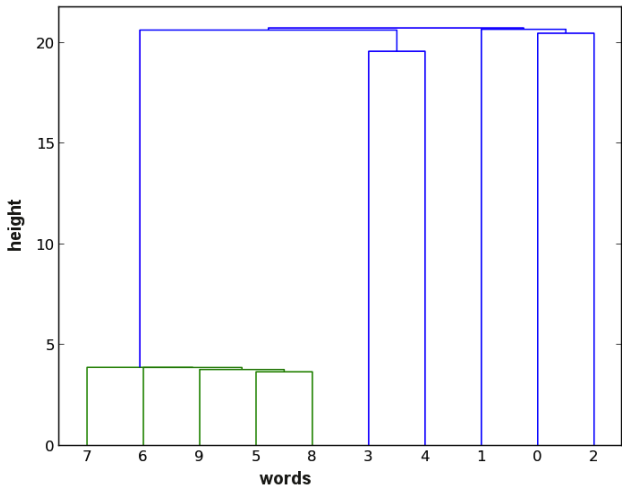


Figure 2: A sample dendrogram resulting from hierarchical clustering.

"horse"	"stationery"	"food"	"party"
horse	paper	apple	dance
eye	paint	fish	protest
farm	plate	bread	music
	card	cake	party
	metal		door
			sound
			weather
			wind

Table 1: Example: Feature-Based Clustering

Agglomerative algorithms are bottom-up in nature. GAAC consists of the following steps:

1. Compute proximity matrix. Each data item is an initial cluster.
2. From the proximity matrix, form pair of clusters by merging. Update proximity matrix to reflect merges.
3. Repeat until all clusters are merged.

A sample dendrogram is shown in Figure 2. The dendogram is pruned at a height depending on the number of desired clusters. The *group average* between the clusters is given by the average similarity distance between the groups. Distances between two clusters and similarity measures are given by the equations below:

$$X_{sum} = \sum_{d_m \in \omega_i \upsilon \omega_i} \sum_{d_n \in \omega_i \upsilon \omega_j d_n \neq d_m} \overrightarrow{d_n}.\overrightarrow{d_m} \qquad (4)$$

$$sim\,(\omega_i, \omega_j) = \frac{1}{(N_i + N_j)\,(N_i + N_j - 1)} X_{sum} \qquad (5)$$

where $\overrightarrow{d}$ is the vector of document of length d. Vector entries are boolean, 1 if the feature is present, 0 otherwise. N_i, N_j is the number of features in ω_i and ω_j respectively, which denote clusters.

The main drawback of the hierarchical clustering algorithm is the running complexity [3], which averages $\theta(N^2 log\ N)$.

We choose average link clustering because our clustering is connectivity-based. The concept proximity matrix consists of features from the knowledge base and 'good' connections occur when two concepts share multiple features.

After clustering, the number of clusters are determined and the dendrogram is pruned accordingly. The output of this process is the set of topics present in the document.

Table 1 provides an example of the results of feature-based clustering.

Topic	Latent Dirichlet Allocation			Commonsense with GAAC		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1	0.67	0.2	0.31	0.875	0.7	0.78
2	0.58	0.67	0.62	0.86	0.67	0.75
3	0.3	0.2	0.24	1.0	0.98	0.99

Table 2: Results for *News* article

Topic	Latent Dirichlet Allocation			Commonsense with GAAC		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1	0.25	0.2	0.22	1.0	0.625	0.77
2	0.12	0.4	0.18	0.27	0.8	0.4
3	0.2	0.31	0.24	0.58	0.85	0.69

Table 3: Results for *Religion* article

5. EXPERIMENTAL RESULTS

5.1 Evaluation of Cluster Quality

Evaluating topic modeling algorithms is not straight-forward. Evaluation measures should seek to compare apples to apples, yet existing algorithms for topic modeling are all either statistical or probabilistic in nature. Therefore, we rely on standard measures (*precision*, *recall* and *F-measure*) [36] to evaluate our model.

These measures are defined as follows:

$$\text{recall} = \frac{TP}{TP + FN} \qquad \text{precision} = \frac{TP}{TP + FP}$$

$$\text{TP rate} = \text{recall}$$

$$\text{FP rate} = \frac{FP}{FP + TN}$$

$$\text{F-measure} = 2 * \frac{precision * recall}{precision + recall}$$

For the probabilistic models, precision and recall are calculated based on the topics in which words occur with the highest probability.

5.2 Experimental Setup

Our testing dataset was derived from the Brown Corpus. Two 300-word test sets, each with three separate topics, were extracted from the Brown categories *Religion* and *News*. Religion topics included 'Nation/Country', 'Christianity', and 'Economy/Politics', and news topics included 'body/injury', 'game', and 'places'. The number of clusters was set to 7 for evaluation purposes. Given that LDA provided the best performance of all pre-existing topic modeling algorithms, we used it as the baseline for comparison. Results are shown in Table 2.

5.3 Discussion

It is difficult to compare models that rely on training, such as LDA, and those that do not, including the method proposed here. Firstly, the large amount of knowledge available to our method could be seen as an unfair advantage.

Secondly, it is difficult to determine a training set that would be comparable to the knowledge available in our commonsense database. Beyond this, short documents present issues for LDA as they do not provide sufficient text to make co-occurrence a useful metric.

For any statistical text mining algorithm, test sets should be independent of the training data but follow the same probability distribution as that data. Our algorithm does not consider probability distributions, and thus is independent of any such requirement.

In order to attempt to overcome these issues, training of the LDA model was attempted with the entire Brown corpus, as training with the entire commonsense knowledge base would have resulted in an overwhelmingly large number of potential topics. Trained LDA did not perform well, however, because relevant words in the test document were not likely to appear in the topic list given that their relative level of co-occurrence for each topic in the corpus was quite low. For comparison, an HMM-LDA model [16] (trigram) was also tested, but this model suffered from the same problems (although at a larger scale due to the inclusion of bigrams and trigrams). Thus, untrained LDA was used for comparison in Tables 2 and 3.

6. CONCLUSION

We have presented a commonsense-based topic modeling algorithm using INTELNET 'semantic atoms' and clustering to determine the topics present in particular documents. We also provide evidence of the algorithm's improved performance over the state-of-the-art algorithm.

One key future extension of this work is to use *language modeling* in place of the current semantic parsing algorithm. In particular, we plan to use topic knowledge in conjunction with the COGPARSE [31] semantic parsing engine to identify semantic and syntactic patterns based on topic information, driving a new approach to language modeling.

This would allow the narrowing of searches to specific paragraphs, taking word order into account and allowing highly targeted searches at sub-document scopes.

7. REFERENCES

[1] S. Arora, R. Ge, and A. Moitra. Learning topic models–going beyond svd. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE, 2012.

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

[3] P. Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[4] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[5] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[6] D. M. Blei and P. J. Moreno. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348. ACM, 2001.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

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

[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

[10] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, volume 2, pages 3–3, 2010.

[11] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society, Series B*, 39(1):1–38, 1977.

[12] N. Eagle, P. Singh, and A. Pentland. Common sense conversations: understanding casual conversation using a common sense database. In *Proceedings of the Artificial Intelligence, Information Access, and Mobile Computing Workshop (IJCAI 2003)*, 2003.

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

[14] 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.

[15] T. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th annual conference of the cognitive science society*, pages 381–386. Citeseer, 2002.

[16] T. Griffiths, M. Steyvers, D. M. Blei, and J. Tenenbaum. *Integrating topics and syntax*, volume 17 of *Advances in Neural Information Processing Systems*, pages 537–544. MIT Press, Cambridge, MA, 2005.

[17] J. Hartigan and M. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society*, 28(1):100–108, 1979.

[18] M. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 23:856–864, 2010.

[19] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[20] N. Howard and E. Cambria. Intention awareness: Improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences*, 3(9), 2013.

[21] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[22] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 61–68, New York, NY, USA, 2009. ACM.

[23] H. Lieberman, H. Liu, P. Singh, and B. Barry. Beating common sense into interactive applications. *AI Magazine*, 25(4):63, 2004.

[24] H. Liu, H. Lieberman, and T. Selker. A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125–132. ACM, 2003.

[25] H. Liu and P. Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[26] 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.

[27] G. Maskeri, S. Sarkar, and K. Heafield. Mining business topics in source code using latent dirichlet allocation. In *Proceedings of the 1st India software engineering conference*, pages 113–120. ACM, 2008.

[28] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[29] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed inference for latent dirichlet allocation. *Advances in neural information processing systems*, 20(1081-1088):17–24, 2007.

[30] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[31] D. Olsher. COGPARSE: Brain-inspired knowledge-driven full semantics parsing: Radical construction grammar, categories, knowledge-based parsing & representation. In *Advances in Brain Inspired Cognitive Systems*, volume 7366 of *LNCS*, pages 1–11. Springer, 2012.

[32] D. Olsher. COGVIEW & INTELNET: Nuanced energy-based knowledge representation and integrated cognitive-conceptual framework for realistic culture,

values, and concept-affected systems simulation. In *Proceedings, 2013 IEEE Symposium Series on Computational Intelligence*, 2013.

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

[34] D. Rajagopal, E. Cambria, D. Olsher, and K. Kwok. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *WWW*, pages 565–570, Rio De Janeiro, 2013.

[35] 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.

[36] M. Steinbach, G. Karypis, V. Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[37] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

[38] Q. Wang, E. Cambria, C. Liu, and A. Hussain. Common sense knowledge for handwritten chinese recognition. *Cognitive Computation*, 5(2):234–242, 2013.

[39] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.