A proposal of the marriage of Encyclopedic and Commonsense Knowledge

Dheeraj Rajagopal
Language Technologies Institute
Carnegie Mellon University
dheeraj@cmu.edu

Niket Tandon
Max Planck Institute for Informatics
Saarbrücken, Germany
ntandon@mpi-inf.mpg.de

Abstract

Commonsense Knowledge (lexical + common knowledge) and Encyclopedic knowledge helps in different situations because one is about entities and other is about named entities. We set up a principled experiment to empirically verify that for real world scenarios, it is best to have a marriage of the types of knowledge. We envision WordNet to be the mediator between these two systems just like Wikipedia is the mediator in linked open data. For this purpose the Yago ontology would be helpful as well.

1 Introduction

There is a growing conviction that the future of machine intelligence will crucially depend on our ability to exploit Big Data for encyclopedic knowledge (for factual knowledge) and commonsense knowledge (for more advanced human-like reasoning).

For encyclopedic knowledge, Knowledge graphs like DBpedia, Freebase, or Yago (Bizer et al., 2009; Bollacker et al., 2008; Suchanek et al., 2007) have become major assets for enriching Web contents and user inputs towards more semantic search and recommendations. Such knowledge is increasingly used at major companies like Facebook, Baidu, Google and Microsoft, among others. The emphasis in these Knowledge graphs is on individual entities like people, organizations, and products with focus on *factual knowledge* about such entities (e.g., songs and awards of an artist, CEOs and products of companies, cities and restaurants visited by friends, etc.).

For commonsense knowledge, Lexical Knowledge acquisition projects like WordNet (manual) and Commonsense Knowledge acquisition projects like Cyc (manual), ConceptNet (semi-automated) and WebChild, SenticNet (automated) have become major assets for understanding and reasoning over human input. Such knowledge is increasingly finding applications ranging from text to vision (Liu and Singh, 2004; Aditya et al., 2015). The emphasis in these knowledge bases is on the classes of entities like living and non-living things and their real-world scenarios with focus on *commonsense knowledge* about such entities (e.g., paper is used in gift wrapping , gift is given on an accomplishment, people become happy on receiving gifts, etc.).

Encyclopedic and Commonsense KBs have existed in isolation and serve isolated applications because either one comprises different genres of knowledge. However, it is unclear whether a combination of the two would be complementary and helpful or not; and if they are found to be complementary, whether there is a gap in the linkage between Encyclopedic and Commonsense KBs. In this paper, we investigate this question by setting up an empirical evaluation for two common real-world tasks in a clustering (Song et al., 2011) scenario.

At the input side of clustering, richer co-occurrences improve the clustering accuracy (Färber et al., 2010). Our first task is to provide richer co-occurrences between the input concepts (using KBs). This additionally entails including concepts that are not present in the input altogether (using KBs). What kind of input can be enriched by which KB and why? We answer this question by analyzing the results of the first task over real-world dataset (Delicious and Flickr) [1] [2]

At the output side of clustering, richer cluster labels improve the interpretation of the output (Role and Nadif, 2014a). Thus, our second task is to provide richer cluster labels as a distribution over concepts (using KBs). This additionally entails including concepts that are not present in the cluster altogether (using KBs). What kind of clusters labels can be provided by which KB and why? We answer this question by analyzing the results of this second task over real-world dataset (Delicious and Flickr).

Finally, we analyze the results of the two tasks to find the answer to an important question, does it help to use a combination of the two genres of knowledge and why? Our answer suggests that a combination of the two genres of knowledge provides the best results. Based on this, we provide a vision of linking the two by proposing a potential mediator, WordNet (Fellbaum, 1998). In this, the Yago ontology that is already linked to WordNet (Suchanek et al., 2007), would be helpful.

In summary, the contributions of this paper are:

- We provide a principled experiment (see Sections 3, 4) to empirically verify (see Section 5) that a marriage of the Encyclopedic and Commonsense Knowledge would help the most in real-world scenarios (see Section 6).
- We envision how such a marriage can be materialized via WordNet (see Section 6).

2 Overview

Figure 1 provides an overview of our system for the two tasks. At the input++ side (task 1), we use two different KBs: Commonsense Knowledge Base (CKB) and Encyclopedic Knowledge Base (EKB). Each KB is independently able to enrich the input to input++. This input++ goes through the -clustering module. The output of this module produces clusters with a distribution over concepts. The last block in the figure, i.e. output++ enhances the clusters by providing them labels based on the knowledge from CKB and EKB. This is the second task. The figure provides an example below each module.

[1] http://www.zubiaga.org/datasets/socialbm0311/

[2] http://press.liacs.nl/mirflickr/

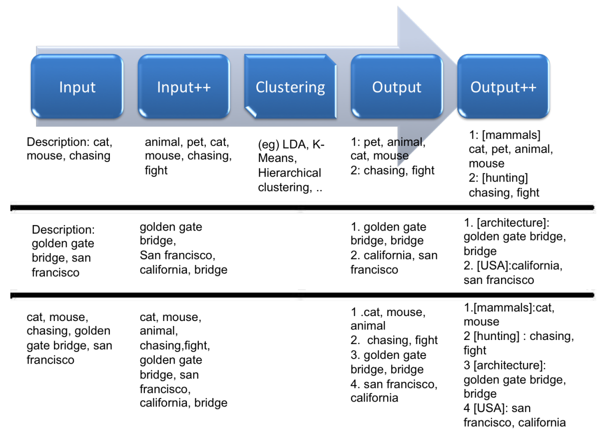

Figure 1: System overview. The figure shows sample output for three different examples, first example input being "A cat is chasing a mouse", second input is "Golden Gate Bridge, San Francisco" and the third corresponds to "A cat chasing a mouse near the Golden Gate Bridge, San Francisco"

3 Task-1: Input++

3.1 Problem Statement

Given an input text document, find a scored list of related concepts from a KB. These related words may be absent in the input document. If we were to cluster the words within a document, the set of words becomes the input to the clustering algorithm. Any clustering algorithm (hard or soft clustering) relies on co-occurrences for clustering (when the document is smaller, the co-occurrences will become very sparse). The expanded set of related words constitutes richer co-occurrences (that we call input++) to a clustering algorithm.

3.2 Approach

For each word w in a document d, our task is to find a ranked list $E(w, KB)$ of related concepts $N(w, KB)$ in a KB. These related concepts may be absent in d i.e. $E(w, KB) \notin W(d)$. As an external source of knowledge, we use a KB (CKB, EKB or a combination of CKB and EKB). First, w is projected in the KB space using a function $F(w, KB)$. On the projected space, we consider the neighborhood of $F(w, KB)$ provided by the function $N(w, KB)$. Finally, $E(w, KB)$ constitutes a subset of the neighborhood candidates.

$F(w, KB)$ projects a string concept to a KB concept, where $w \in W(c)$. A string concept like *Kashmir* could link to multiple KB concepts *song: Kashmir* or *state: Kashmir*. We need to disambiguate the correct sense of w w.r.t. KB. Standard disambiguation techniques (Yosef et al., 2011) exploit the neighborhood of a word (a node is defined by the company it keeps). In our setting, $W(c)$ is the natural neighborhood of w. Thus, we exploit the coherence within cluster members in order to disambiguate the concept. One the one hand we have w with its neighborhood $W(c)$, and, on the other hand we have the k senses of w, $kb(w) \in KB(w)$, based on the surface level string comparison with w. We select the $kb(w)$ whose context (usually given by the $N(w, KB)$) best overlaps with the context $W(c)$ of w. $F(w, KB) = argmax_{kb(w) \in KB(w)} W(c) \cdot kb(w)$. We consider only the best overlapping concept and in case of a tie, we take the more popular KB node i.e. one with the higher degree.

$N(w, KB)$ defines the weighted neighborhood of $kb(w)$. The set of nodes in the KB graph that are directly reachable at a distance of one (via any relation) from $kb(w)$ constitute the neighborhood of $kb(w)$. The weight of a neighbor is the normalized edge weight connecting $kb(w)$ and the neighbor. We represent the neighbor as $N(w, KB)$ and the corresponding weight as $N(w, KB).weight$.

In order to obtain a scored list of external co-occurrences for a word w, we define a matrix M of document's words as rows and a union of their neighborhood as columns. An entry $M(i, j)$ corresponding to the the row i denoting $w \in W(d)$ and column j denoting one of its neighbors $N(w, KB)$ contains the score of $N(w, KB), \geq 0$. The input++ consists of all $j \in cols(M)$ that satisfy the constraint $1 \leq \sum_i M(i, j) \leq \sum_i$. The constraint avoids candidates with low support and noisy hubs that add no particular value because they co-occur with every concept.

For example, lets say a document about "Fruits and its Nutritional Values" contains the following words in their description, *apple, orange, grape, banana, nutrition, vitamins, carbohydrate, calories, breakfast, dessert*. A sample expanded list of words ($\delta = 2$) after the normalization would be ("food", 7), ("fruit", 4") , ("health", 4) , ("eat", 3) , ("nutrition",3), ("fitness",2) ("wellbeing",2). These words are used to enhance the co-occurrences and together with the input, they serve as the input++ for clustering.

4 Task-2: Output++

4.1 Problem Statement

We define descriptive, human-readable labels for the clusters produced by the clustering algorithm. These labels are not produced by the clustering algorithms and can be absent in the cluster altogether. We examine the cluster members to find the labeling that summarizes the topic of each cluster.

4.2 Approach

4.2.1 Constructing and cleaning the cluster

Given the document $\mathcal{D} = (w_1, w_2, .., w_n) + (w_{n+1}, w_{n+2}, .., w_{n+m})$ where $(w_{n+1}, w_{n+2}, .., w_{n+m})$ are the expanded concepts (obtained as a result of input++). The purpose of the clustering algorithm is to group the words in hard or soft clusters.

Selecting every word from the cluster as a representative for the labels might not be effective since their importance measure to the cluster might vary (Treeratpituk and Callan, 2006). Given a cluster $c \in C$, and the words in the cluster $W(c) = (w_1, w_2, ..., w_n)$, our first task is to rank $(w_1, w_2, ..., w_n)$ in the order of their representativeness to the cluster. If the clustering method produces soft clusters already, then we prune out cluster members below a certain threshold. If the clustering method produces hard clusters, then we need to find representative members as described in the next paragraph.

We need representative members from a cluster (in order to prune noisy cluster members). We use a Random walk with restarts model (Tong et al., 2006) in order to find a ranked list of cluster representatives. A frequently visited node during a random walk often tends to be influential among all clusters but not necessarily representative of any cluster (Role and Nadif, 2014b). In order to find representative nodes after a random walk, we augment the resultant score (from stationary distribution obtained after random walk) with a centrality measure (Newman, 2005). We define *centrality* of a node n based on the ratio of the number of outgoing links to the number of inlinks i.e. $centrality(n) = \frac{\#outlinks}{\#inlinks}$. The strong locality in the random walk is amenable to parallelization. This makes the algorithm scalable. Upon convergence of the random walk, we get a stationary distribution. We select a cluster member if the probability of reaching that node via NO_RELATION is greater than the probability of reaching that node via other nodes $\sum_i (centrality_i / num_{out})^{k1} num_{out}^{k2}$.

4.2.2 Labeling the clusters

Given a cleaned clusters $c \in C$, our task is to find a ranked list of labels $L(c)$ that are suitable for the cluster. These labels may be absent in the cluster i.e. $l(c) \notin W(c)$ where, $l(c) \in L(c)$. As an external source of knowledge, we use a KB (CKB, EKB or a combination of CKB and EKB). Each word $w \in W(c)$ is projected in the KB space using a function $F(w, KB)$. On the projected space, we consider the neighborhood of $F(w, KB)$ provided by the function $N(w, KB)$. Finally, $l(c)$ constitutes the neighbors of the cluster members in the KB space that most cluster members attract.

In order to obtain a scored list of labels for a cluster c, we define a matrix M of cluster members as rows and a union of their neighborhoods as columns. An entry $M(i, j)$ corresponding to the the row i denoting $w \in W(c)$ and column j as one of its neighbors $N(w, KB)$ contains the score of $N(w, KB).weight, \geq 0$. The scored list $l(c)$ consists of all $j \in cols(M)$ whose $\cup_j \sum_i M(i, j) \geq \delta$. δ is a manually defined threshold.

For instance, a cluster containing the following concepts *'bite' ,'chewy', 'pretty', 'butter', 'bar'* would be labelled as *("food",4) and ("eat",3)*. From the resulting distribution of concepts, we select a percentile range to filter out the noise of the frequently occurring concepts that are noisy. Concepts like *fun*, *work*, *object*, *verb* and *concept* occur across many clusters but they are not specific enough to associate identity to a particular cluster.

5 Results

5.1 Dataset

For experimental purposes, we use the social tagging dataset from Delicious and Flickr. We choose the Delicious and Flikr dataset and the tag recommendation problem since they are natural to the setting of input++ and output++. If we consider the webpage of the bookmark(Delicious) and the image described(Flickr) to be latent, then the descriptions are analogous to input(enriched input to input++), and after clustering, output++ become analogous to the tags. This makes them the ideal choice for evaluating the system.

We used the Social-ODP-2K9-dataset (Zubiaga et al., 2009) for Delicious bookmarks. The dataset consisted of 12,616 unique URLs with a weighted list of tags, the bookmarks and the notes from Delicious. For Flickr, we used the MediaMill Tag Relevance dataset (Li et al., 2009), a collection of 3.5 million images and their respective tags from Flickr.

For our experiments, we used a random sample of 1000 Delicious bookmarks and a random sample of 8000 Flickr image-tags-description. We assume that the individual KBs will have enough knowledge to cover such a wide-range of words in both the corpora.

5.2 Knowledge Bases

CKB: Our commonsense knowledge base (CKB) encompasses a combination of lexical and common knowledge base. For common knowledge, we use the COGBASE system (Olsher, 2014). COGBASE includes the lexical knowledge from WordNet.

EKB: YAGO is one of the most widely used encyclopedic knowledge bases containing mainly data from Wikipedia and WordNet.

5.3 Evaluation Metrics

To evaluate input++ and output++ , we use the *Jaccard Index* and *WordNet Path Similarity* (Leacock and Chodorow, 1998) measure to evaluate the quality of the enhanced input and output. We examine with experiments how different knowledge bases perform better with different settings. The Jaccard Index measures the similarity of the input(++) to the output(++). The jaccard similarity is given as

$$Jaccard\_Index = \frac{A \cap B}{A \cup B} \tag{1}$$

To find the relevance of the input++ extracted from different KB, we use the WordNet Path similarity measure. *Path Similarity* measures the distance between two concepts in the WN taxonomy.

The Path Similarity between two synsets C_1 and C_2 is given by

$$Sim_{path}\left(C_1, C_2\right) = 2 * deep\_max - len\left(C_1, C_2\right)$$

where $deep\_max$ is the $maxdepth(c_i)$ of the taxonomy and $len(C_i, C_j)$ is the length of the shortest path from synset C_1 to C_2 in WordNet.

5.4 Evaluation

We first evaluate the strengths of individual KBs on the Delicious and Flickr datasets. Table 1 shows the baselines for input++, output++, and performance of CKB and EKB over input++ and output++. For clustering, we use a commonsense based hard-clustering algorithm as described in (Rajagopal et al., 2013).

Source		δ		
Input	Output	1	2	3
Input	Output			0.07
Input	EKB	0.06	0.06	0.08
Input	CKB	0.05	0.07	0.07
CKB	Output	0.03	0.06	0.07
EKB	Output	0.05	0.06	0.07
CKB	EKB	0.07	0.07	0.08
CKB	CKB	0.13	0.09	0.08
EKB	EKB	**0.18**	**0.13**	**0.11**
EKB	CKB	0.07	0.07	0.07

Table 1: Jaccard Similarity Results for individual KBs on Delicious dataset

It is evident from the results that EKB not only provides richer co-occurrence but also provides the the best tags for the Delicious setting and it outperforms the baseline and CKB by a considerable margin. But, we observed contrasting results in the Flickr setting. Table 2 shows the baseline and individual KB performance for input++ and output++. CKB showed greater promise when it comes to having input++ and output++ for Flickr.

Our second evaluation strategy for input++, is using the *WordNet PathSimilarity* (WNPathSim) measure. For each input++, we take the sum of the WNPathSim between the original input and the relevant words extracted from the KB(lower the distance, higher the relevance). This result gives us the much required insight about choosing the optimal δ

Source		δ		
Input	Output	1	2	3
Input	Output			0.09
Input	EKB	0.03	0.05	0.06
Input	CKB	0.03	0.07	0.07
CKB	Output	0.03	0.07	0.07
EKB	Output	0.03	0.05	0.07
CKB	EKB	0.03	0.06	0.11
CKB	CKB	0.11	**0.14**	**0.18**
EKB	EKB	**0.13**	0.10	0.11
EKB	CKB	0.12	0.10	0.12

Table 2: Jaccard Similarity Results for individual KBs on Flickr dataset

to select input++ for the clustering algorithm. Figures 2 and 3 shows the trend of relevance of input++ at each δ. At δ = 3, the WNPathSim measure for input++ for individual KBs are at the closest to the original input and hence they are more likely to be relevant.

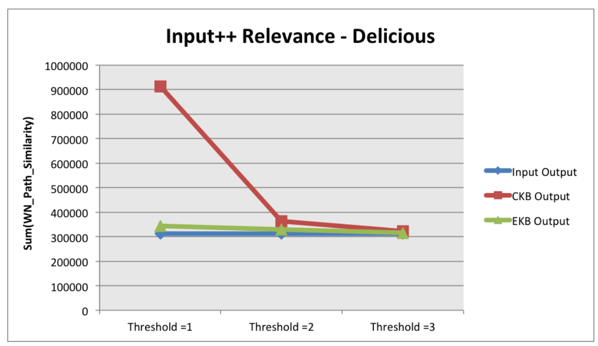

Figure 2: Input++ Relevance Comparison for Delicious

From the results, we observe that no single KB can provide better better input co-occurrence and cluster labels across different datasets. Figure 4 shows the performance of KBs across datasets(for δ = 3). Section 6 examines the effectiveness of a combined approach between EKB and CKB.

6 Analysis

6.1 Would a marriage of CKB and EKB be fruitful?

If CKB and EKB perform better at different datasets, would a combination of both enhance the overall

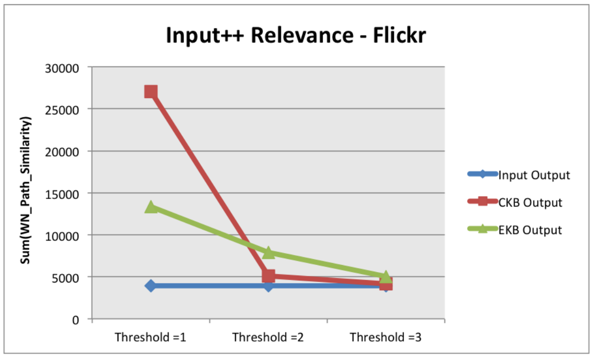

Figure 3: Input++ Relevance Comparison for Flickr

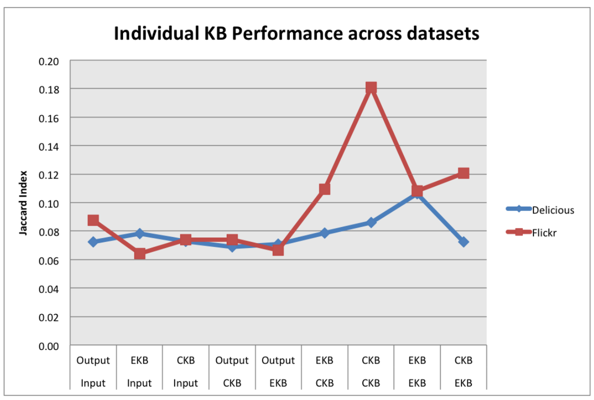

Figure 4: Comparative Evaluation for Delicious and Flickr datasets at δ = 3. On the x-axis, the top row denotes the output and the bottom row the input

performance across datasets? We answer this question by evaluating the marriage of CKB and EKB, we examine how the combination of CKB+EKBs perform the same task under the same experimental settings as the individual KBs. The results in the table 3 and 4 shows the results of such a marriage.

Figure 5 shows the performance of the KB combination at δ = 3. We observe a trend in the Delicious dataset, where the combination of CKB and EKB outperforms the individual KBs whereas in the Flickr setting, the combination performs slightly lower than the CKB. Overall, the combination EKB and CKB for input++ and output++ performs best across both the datasets (shown in figure 6).

Source		δ		
Input	Output	1	2	3
Input	CKB+EKB	0.05	0.06	0.08
CKB	CKB+EKB	0.14	0.09	0.09
EKB	CKB+EKB	0.14	0.13	0.11
CKB+EKB	Output	0.02	0.05	0.07
CKB+EKB	CKB	0.13	0.08	0.08
CKB+EKB	EKB	0.14	0.13	0.11
CKB+EKB	CKB+EKB	**0.20**	**0.15**	**0.12**

Table 3: Jaccard Similarity Results for CKB + EKB on Delicious dataset

Source		δ		
Input	Output	1	2	3
Input	CKB+EKB	0.02	0.04	0.06
CKB	CKB+EKB	0.08	0.10	0.15
EKB	CKB+EKB	0.08	0.10	0.12
CKB+EKB	Output	0.02	0.04	0.06
CKB+EKB	CKB	0.09	0.10	0.15
CKB+EKB	EKB	0.08	0.10	0.12
CKB+EKB	CKB+EKB	**0.12**	**0.12**	**0.16**

Table 4: Jaccard Similarity Results for CKB + EKB on Flickr dataset

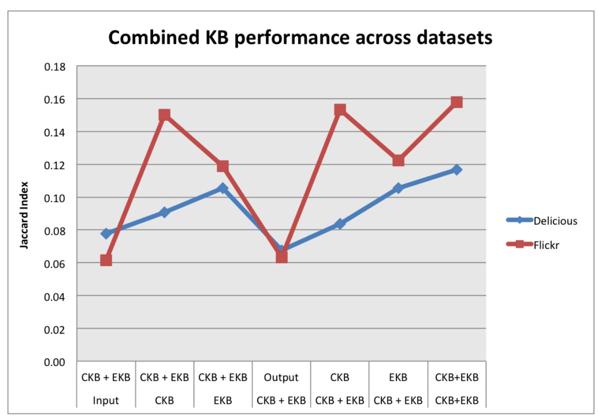

Figure 5: CKB+EKB Performace for Delicious and Flickr datasets at δ = 3. On the x-axis, the top row denotes the output and the bottom row the input

6.2 Realizing a marriage of CKB and EKB

Realizing such a marriage requires some mediation so that the KBs can communicate. The concepts from one KB has to be correctly mapped to the other

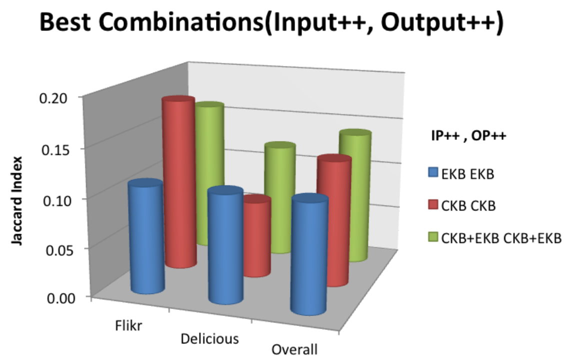

Figure 6: Best performing combinations for Flickr and Delicious at δ = 3

in order to connect them. We use WordNet as our mediator for this system between EKB and CKB.

7 Related Work

Unsupervised Social Tagging systems rely on external knowledge bases to suggest tags. There have been several approaches that examine the use of knowledge bases in cluster labeling. The method used in (Syed et al., 2008) and (Carmel et al., 2009) explores the ability of using wikipedia as the external knowledge base for labeling clusters. The approach used by (Tseng, 2010) and (Bouras and Tsogkas, 2012) use wordnet as their external resource and labels are generated using hypernym relations of the cluster candidates.

Different knowledge bases focus on extracting different types of knowledge. We believe that natural language texts use a combination of knowledge to express the meaning. Our work examines the strengths and weaknesses of using different knowledge bases and it aligns best with encyclopedic and commonsense knowledge bases.

8 Conclusion

We explore the possibility of using different types of knowledge bases for cluster labeling and their advantages and disadvantages of using them in tandem relative to the dataset they are applied to. Our results suggest that using combination of KBs enables us to predict better cluster labels as compared to using individual KBs. This is very much in resonance with our hypothesis that humans use multitude of knowledge in natural language. For our future work, we want to explore stronger models to combine KBs and extend the current work to multi-lingual text.

References

Somak Aditya, Yezhou Yang, Chitta Baral, Cornelia Fermuller, and Yiannis Aloimonos. 2015. Visual commonsense for scene understanding using perception, semantic parsing and reasoning. In *2015 AAAI Spring Symposium Series*.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Christos Bouras and Vassilis Tsogkas. 2012. A clustering technique for news articles using wordnet. *Knowledge-Based Systems*, 36:115–128.

David Carmel, Haggai Roitman, and Naama Zwerdling. 2009. Enhancing cluster labeling using wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 139–146. ACM.

Ines Färber, Stephan Günnemann, Hans-Peter Kriegel, Peer Kröger, Emmanuel Müller, Erich Schubert, Thomas Seidl, and Arthur Zimek. 2010. On using class-labels in evaluation of clusterings. In *MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD*, page 1.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

Xirong Li, Cees GM Snoek, and Marcel Worring. 2009. Learning social tag relevance by neighbor voting. *Multimedia, IEEE Transactions on*, 11(7):1310–1322.

Hugo Liu and Push Singh. 2004. Commonsense reasoning in and over natural language. In *Knowledge-based intelligent information and engineering systems*, pages 293–306. Springer.

Mark EJ Newman. 2005. A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54.

Daniel Olsher. 2014. Semantically-based priors and nuanced knowledge core for big data, social ai, and language understanding. *Neural Networks*.

Dheeraj Rajagopal, Daniel Olsher, Erik Cambria, and Kenneth Kwok. 2013. Commonsense-based topic modeling. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 6. ACM.

François Role and Mohamed Nadif. 2014a. Beyond cluster labeling: Semantic interpretation of clusters' contents using a graph representation. *Know.-Based Syst.*, 56:141–155, January.

François Role and Mohamed Nadif. 2014b. Beyond cluster labeling: Semantic interpretation of clusters contents using a graph representation. *Knowledge-Based Systems*, 56:141–155.

Yang Song, Lu Zhang, and C Lee Giles. 2011. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1):4.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Zareen Saba Syed, Tim Finin, and Anupam Joshi. 2008. Wikipedia as an ontology for describing documents. In *ICWSM*.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications.

Pucktada Treeratpituk and Jamie Callan. 2006. Automatically labeling hierarchical clusters. In *Proceedings of the 2006 international conference on Digital government research*, pages 167–176. Digital Government Society of North America.

Yuen-Hsien Tseng. 2010. Generic title labeling for clustered documents. *Expert Systems with Applications*, 37(3):2247–2254.

Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. 2011. Aida: An online tool for accurate disambiguation of named entities in text and tables. *Proceedings of the VLDB Endowment*, 4(12):1450–1453.

Arkaitz Zubiaga, Raquel Martínez, and Víctor Fresno. 2009. Getting the most out of social annotations for web page classification. In *Proceedings of the 9th ACM symposium on Document engineering*, pages 74–83. ACM.