A Graph-Based Approach to Topic Clustering of Tourist Attraction Reviews

Nuttha Sirilertworakul[1(✉)] and Boonsit Yimwadsana[1,2]

[1] Faculty of Information and Communication Technology, Mahidol University, 999 Phuttamonthon 4 Road, Salaya 73170, Nakhon Pathom, Thailand
nuttha.sii@student.mahidol.ac.th, boonsit.yim@mahidol.ac.th

[2] Integrated Computational BioScience Center, Office of the President, Mahidol University, 999 Phuttamonthon 4 Road, Salaya 73170, Nakhon Pathom, Thailand

Abstract. A large volume of user reviews on tourist attractions can prohibit travel businesses from acquiring overall consumers' expectations and consumers themselves from seeing the big picture and making thoughtful decisions on trip planning. Summarization of the reviews allows both parties to catch the main themes and underlying tones of the attractions. In this paper, we address the task of topic clustering, by applying a graph-based approach to group the reviews into clusters. To interpret the resulting review clusters, WordNet and Inverse Document Frequency (IDF) are utilized to extract keywords from each cluster which represents the topic. We evaluate the graph-based clustering approach against gold standard data annotated by human and the results are compared against Latent Dirichlet Allocation (LDA), a widely used algorithm for topic discovery. The approach is shown to be competitive to LDA in terms of clustering user reviews on tourist attractions. The graph-based approach, unlike LDA which requires the number of clusters as an input, can dynamically clusters the reviews into groups, revealing the number of clusters.

Keywords: Text mining · Text summarization · Topic clustering · Graph clustering · User reviews

1 Introduction

User reviews on tourist attractions provide travel businesses and consumers with a mixture of fact and opinion which are significant to their decision making [15]. While businesses require user reviews for product and service improvement, consumers need them for trip planning. Summarizing the reviews allows both parties to catch the main themes and underlying tones of the attractions.

Summarization can be performed by discovering patterns of words in a collection of documents which reflect its underlying topics [2]. Not only topic discovery reveals general ideas of the documents, there are several applications based on it. Some examples are topic-based searching [9], contextual advertising [17], and recommender systems [12].

R. Damaševičius and G. Vasiljevienė (Eds.): ICIST 2019, CCIS 1078, pp. 343–354, 2019.
https://doi.org/10.1007/978-3-030-30275-7_26

In this paper, we apply Soft-Regularized Markov Clustering (SR-MCL) [20] to cluster user reviews and extract keywords from each resulting cluster which represents the topic, through the utilization of WordNet[1] and Inverse Document Frequency (IDF). The advantage of the graph-based approach over Latent Dirichlet Allocation (LDA) [5] is its ability to dynamically group reviews into clusters and thus provide the number of clusters which is usually unknown and varies depending on the nature of the data.

2 Related Work

2.1 Text Summarization

Text summarization is the task of "grouping together of similar information and describing those groups" [16]. The resulting groups can be described using either *extractive* or *abstractive* approach. Extractive summarization uses units of text, such as words and sentences, from data in the group to represent the group while abstractive summarization represents the group with a new description, similar to the approach taken by human.

As described in [16], we explore extractive summarization which includes two main tasks: (1) topic clustering which groups similar reviews into clusters and (2) keyword extraction which draws keywords from the resulting clusters to represent the topics. In this paper, the focus is on the task of topic clustering and its evaluation against LDA clustering. For the task of keyword extraction, we investigate the utilization of WordNet and IDF to semantically draw meaningful words from each cluster, representing a topic.

2.2 Markov Clustering

Soft-Regularized Markov Clustering (SR-MCL) [20] is a graph clustering algorithm which is a variant of Markov Clustering (MCL) [7] and Regularized Markov Clustering (R-MCL) [18]. MCL runs two operations, *Expand* and *Inflate*, iteratively on a stochastic matrix, M, which describes the transition probabilities from each of the nodes in the graph to the others. The Expand operation is $M := M \times M$, and the Inflate operation raises each cell in the matrix by the Inflate parameter (r), and re-normalizes the sum of each column to 1, representing the transition probabilities.

R-MCL later modified MCL by replacing the Expand operation with *Regularize* operation which is $M := M \times M_G$, where M_G is a canonical stochastic matrix storing the original transition probabilities of the nodes within the graph. The Regularize and Inflate operations are run iteratively, spreading more flows within the same cluster than between clusters as intuitively there are more paths between nodes within the same cluster. The algorithm is terminated when there is no change in the matrix M or the number of iterations is reached. After the termination, nodes that flow to the same 'attractor node' belong to the same

[1] Lexical database which measures the relatedness of terms.

Algorithm 1. SR-MCL [20]

Input: The canonical stochastic matrix M_G, the balance parameter b, the inflation parameter r, the penalized ratio β and the number of iterations t.
Output: A set of clusters C.

$C = \{\}$
$count = \{0, 0, ..., 0\}$
for $iter \rightarrow t$ **do**
  repeat
    $M_R = RegularizationMatrix(M, M_G, b)$
    $M = M * M_R$ ▷ Regularize
    $M = Inflate(M, r, count, \beta)$ ▷ Inflate
    $M = Prune(M)$
  until M converges ▷ iter-time execution of R-MCL
  $T_{iter} = attractors(M)$ ▷ Resulting attractor nodes from iter-time execution of R-MCL
  for all $v_i \in T_{iter}$ **do**
    $count[i] = count[i] + 1$
  end for
  $C_{iter} = clusters(M)$ ▷ Resulting clusters from iter-time execution of R-MCL
  $C = C \cup C_{iter}$
end for
$C = post\text{-}process...(C)$ ▷ Extract qualified clusters

cluster. Additionally, a balance parameter (b), is introduced in [19] to inhibit the flow to an attractor node which attracts a large number of nodes. This produces the resulting clusters which are more balanced in size.

In SR-MCL, the idea is to "iteratively re-execute R-MCL while ensuring that the resulting clusters are not always the same" [20] by introducing a penalty parameter (β) in the Inflate operation. The parameter decreases the flow to a node which has been an attractor x times in the previous iterations by raising the node (i.e., the corresponding column in the matrix M) to $r \times \beta^x$", where $\beta > 1$. After the algorithm is terminated, post-processing step removes redundant and low-quality clusters according to the quality function. The default quality function is density multiply by the square root of size. The advantage of SR-MCL over MCL and R-MCL is its soft-clustering property which means a node can belong to more than one cluster. The SR-MCL process is shown in Algorithm 1.

Typical applications of MCL and its variants are in clustering biological networks such as protein-protein interaction networks [20]. Few applications are also found in text corpora. In [8], MCL is applied to automatically discover word senses from text. In [13], a technique to classify Arabic documents is proposed through a combination of MCL and Deep Belief Networks. Another closely related work is topic clustering of comments on online news [1] which applied MCL to group comments and the resulting clusters are labeled by incorporating information from DBpedia. One limitation on topic clustering of this approach is that MCL is a hard-clustering algorithm (i.e., each comment is assigned to only one cluster).

This paper applied SR-MCL which is a soft-clustering algorithm to cluster user reviews on tourist attractions, allowing the reviews to belong to more than one cluster. This corresponds with the nature of a review which is usually composed of more than one topic.

3 Dataset

English written reviews from 4 popular tourist attractions in Bangkok, Thailand of different categories are downloaded from TripAdvisor[2]. The four attractions are Lumphini Park (nature), Wat Pho (landmarks), Jim Thomson (museums), and Siam Paragon (shopping). Each attraction contains more than 3000 reviews. All of the reviews in each attraction are used to generate a separate set of review clusters and topics.

To evaluate the resulting clusters, a gold standard dataset is created by human annotators using a similar approach as in [1]. Using a 95% confidence level and ±10% margin of error, a set of 100 random reviews from each of the four attractions are provided to two annotators to group them into clusters by following the steps as: (1) Read the reviews and come up with topics labeled by a word or phrase. For instance, an annotator may come up with topics Food, Shopping, and Transportation when provided with reviews from a shopping attraction. (2) Assign the reviews to one or more created topics (i.e., clusters). On average, an annotator creates 9.31 clusters per attraction.

4 Method

4.1 Document Clustering

Since our approach applied SR-MCL which is a graph algorithm to perform document clustering, a set of reviews from each tourist attraction is represented as a graph. The graph is denoted as $G(V, E, W)$ where V is a set of nodes which represent the reviews and E is a set of edges which connect the nodes (i.e., reviews).

The set of edges, E, has a corresponding set of weights, W. The weight $w_{i,j}$ on an edge $e_{i,j}$ is a cosine similarity value between nodes v_i and v_j. In our approach, there is an edge, $e_{i,j}$, between nodes v_i and v_j if the similarity value between the two nodes is greater than or equal to a similarity threshold, x, ranging from (0, 1).

In practice, the graph is constructed from the reviews on each attraction as: (1) Data preprocessing is performed by tokenizing each review into terms, converting all terms to lowercases, removing non-alphabetic characters and stopwords, and stemming the terms into their root forms. This step generates a bag of words from the input reviews. (2) Feature extraction constructs a term-document matrix (e.g., TF-IDF matrix), which stores the occurrences of each

[2] https://www.tripadvisor.com/.

term within each review. (3) Cosine similarity measure is applied to the term-document matrix to find a similarity score between each pair of reviews, forming a pairwise similarity matrix, S. (4) The adjacency matrix, A, which corresponds to a graph, is computed by pruning the similarity matrix according to the similarity threshold (x) discussed in the previous paragraph:

$$a_{i,j} = \begin{cases} s_{i,j}, & \text{if } s_{i,j} \geq x \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The adjacency matrix is then re-normalized by columns. The result is a stochastic matrix which is parsed to SR-MCL algorithm shown in Algorithm 1. The implementation[3] of SR-MCL used by this work is provided by [6].

4.2 Topic Extraction

SR-MCL returns a set of review clusters. To extract a topic which is represented as keywords from each review cluster, WordNet and Inverse Document Frequency (IDF) are applied.

Since keywords of the topic are semantically similar to other probable words within the cluster, WordNet is used to give more weight to those words [3]. Given $n(w,t)$ be a function which returns the total count of word w from documents in cluster t, C_t be a set of words from documents in cluster t, and $PS(w,v)$ be a function which returns path similarity[4] between the words w and v in WordNet which ranges between [0, 1]. The word count $WN(w,t)$ of word w in topic t is updated as:

$$WN(w,t) = n(w,t) + \sum_{v \neq w \in C_t} n(v,t) \times PS(w,v) \tag{2}$$

Then IDF is applied at the cluster level (i.e., reviews in the same cluster are treated as a single document) to reduce the importance of words which occur in many clusters. Given $IDF(w)$ be a function which returns inverse document frequency of word w. The term-topic score of word w in topic t is calculated by multiplying the probability of the updated word count by IDF as:

$$Term\text{-}Topic(w,t) = \frac{WN(w,t)}{\sum_{v \in C_t} WN(v,t)} \times IDF(w) \tag{3}$$

Lastly, the words in each cluster are arranged by their term-topic scores in descending order which reflects the importance of words within the cluster, representing the topic.

3 https://github.com/koadman/proxigenomics.

4 Path similarity computes the shortest path between two word senses. Word senses are more similar when their path distance is closer to 1.

4.3 Topic Tagging

Each document is tagged with generated topics, together with their probabilities distributed over the document. Let W_t be a set of top-N words from topic t, W_d be a set of words from document d, T_d be a set of topics document d is composed of (i.e., a set of clusters document d belongs to), and $TFIDF(w, d)$ be a function which returns term frequency-inverse document frequency of word w in document d. The probability distribution of topic t on document d is computed:

$$Topic\text{-}Doc(t, d) = \frac{\sum_{w \in W_d \cap W_t} TFIDF(w, d)}{\sum_{k \in T_d}(\sum_{v \in W_d \cap W_k} TFIDF(v, d))} \tag{4}$$

5 Experiments and Results

5.1 Cluster Granularity in SR-MCL

Instead of being supplied with the number of clusters as in LDA, SR-MCL has the Inflate parameter (r) which reflects cluster granularity. Figure 1 shows the BCubed[5] precision, recall, and the number of clusters as we varied r between 1.1 and 2.5. The precision varies slightly around 0.70 while the recall fluctuates strongly when the value of r is higher. A larger value of r also results in finer granularity and thus more clusters are generated.

In the following experiments, r is set to 1.5 as in [1] to demonstrate the robustness of the parameter against different text corpora.

5.2 Comparison with Latent Dirichlet Allocation

The approach is compared against LDA, a widely used algorithm for topic discovery. An LDA model is created for each set of reviews. Since LDA requires a predetermined number of topics, it is set to the average number of clusters [1] within the gold standard dataset mentioned in Sect. 3. The average number of topics is 9.31, so the number of topics is set to 9 topics for each LDA model created. For the LDA hyperparameters β and α, they are set respectively to 0.1 and 50/T, where T is the number of topics as suggested in [10]. Since LDA distributes multiple topics as probabilities over a document, a threshold is used to determine the topics of which a certain document is made. In this work, a threshold is set to be $1/T$, where T is the number of topics. For SR-MCL configurations, r is set to 1.5 as preciously tuned in [1] to work with text corpora. The other parameters are set to default values [20].

For evaluation, the SR-MCL clusters and LDA clusters are compared with the gold standard dataset described in Sect. 3. BCubed precision, recall, and F-measure are used as evaluation metrics since they fulfill the formal cluster constraints: cluster homogeneity, completeness, rag bag, and cluster size versus quantity [4].

[5] BCubed is an evaluation metric which compares the resulting clusters generated by an algorithm with the gold standard clusters.

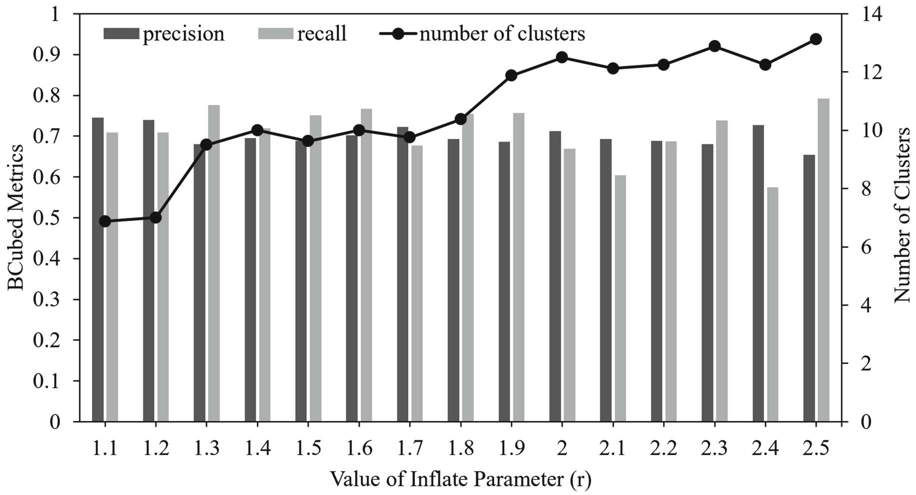

Fig. 1. BCubed precision, recall, and the number of generated clusters when the Inflate parameter is varied.

Using BCubed to estimate the fit between two sets of clusters X and Y at an item level, precision measures a proportion of items sharing a cluster with a particular item i in X which appears in its cluster in Y and recall measures a proportion of items sharing a cluster in Y with the item i which appears in its cluster in X [14]. For F-measure, we use F1 score which is the harmonic mean of precision and recall. The evaluation employed follows the BCubed definition discussed in [4].

The clustering evaluation results are shown in Table 1. The fit between Human1-Human2 clusters, according to BCubed metrics, is around 0.73. To compare the fits of SR-MCL-to-human and LDA-to-human clusters, a paired t-test is performed. According to the test, we cannot reject the hypothesis that SR-MCL clustering performs better than LDA clustering in terms of precision ($p = 0.28$). Yet, the mean recall and F-measure of SR-MCL clustering are significantly[6] higher than those of LDA clustering, with p-values of 0.01 and 0.04 respectively.

Table 1. Average BCubed Precision, Recall, and F-measure metrics.

Metric	Human1-Human2	SR-MCL-Human	LDA-Human
BCubed Precision	0.76	0.79	0.82
BCubed Recall	0.71	0.63	0.48
BCubed F-measure	0.73	0.69	0.60

[6] There is a statistically significant difference between the two results if p-value is less than 0.05 ($p < 0.05$).

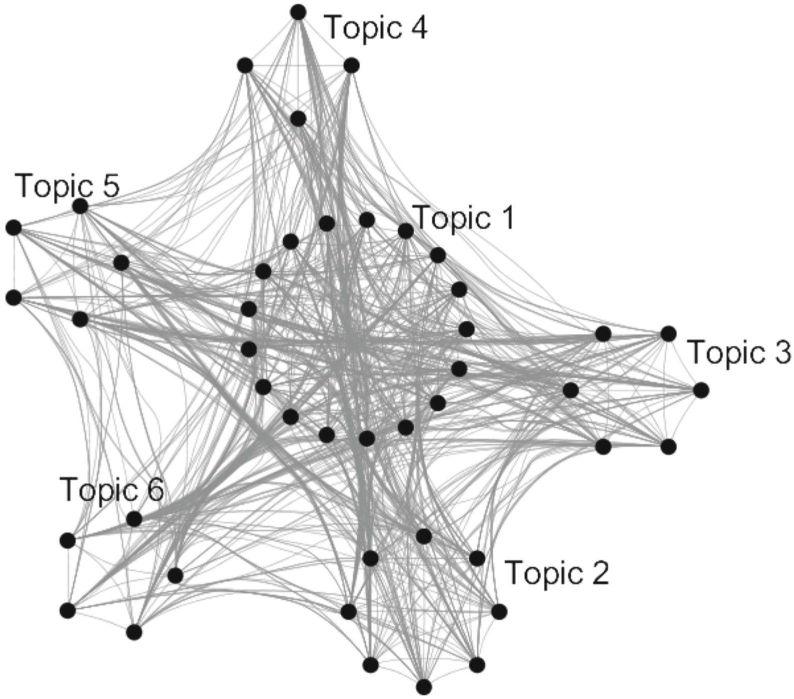

Fig. 2. Document-topic relationship

5.3 Identifying Document-Topic Relationship

Aiming to ease the interpretation of clustering results, we illustrate document clustering as a graph with nodes representing the reviews. Each node has an attribute, most-likely topic, which indicates the topic the review is made up of with the highest probability as computed using Eq. 4. The edges connect the nodes sharing one or more clusters (i.e., topics). The weight on an edge indicates the number of topics shared by the two reviews.

Figure 2 illustrates an example of the graph using Cytoscape[7], a graph visualization tool. The nodes, whose most-likely topics are the same, are circularly grouped together and the edges are bundled using a built-in algorithm invented by [11] to reduce the clutter within the graph and highlight the connectivity strengths between the reviews and the topics.

[7] https://cytoscape.org/.

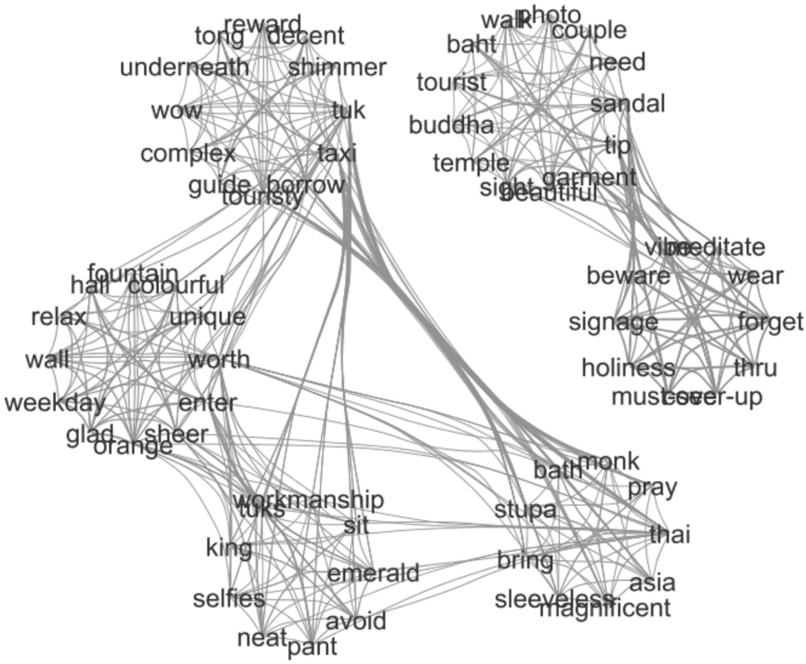

Fig. 3. Word-topic relationship

5.4 Identifying Term-Topic Relationship

Similar to Sect. 5.3 with the aim of investigating the semantics, we present the relationship between words and topics as a graph with nodes representing the words. Each node has most-likely topic as an attribute, stating the topic the word resides and has the highest term-topic score among its scores in other topics. The edges connect the nodes which share one or more topics. The weight on an edge is an average of term-topic score multiplication of the two words co-occurring within the same clusters. Let avg_t be a function which returns an average of the series where t is the topic(s) the two words co-occur and $Term\text{-}Topic(w, t)$ be a function which returns term-topic score of word w in topic t. The weight of edge between word w and word v is computed as:

$$W(w, v) = avg_t(Term\text{-}Topic(w, t) \times Term\text{-}Topic(v, t)) \tag{5}$$

Figure 3 shows an example of the graph with nodes circularly grouped by their most-likely topics and edges bundled, highlighting the connectivity strengths between the words and the topics.

6 Discussion

The results of topic clustering based on SR-MCL demonstrate that the approach is an alternative to LDA to cluster reviews of tourist attractions. Unlike LDA, SR-MCL does not require the number of clusters as prior knowledge. Yet, its Inflate parameter affects the number of clusters. To demonstrate its robustness against different text corpora, we adopted a previously tuned value [1]. The value results in relatively high precision and recall when compared with other values of r.

The limitation of the demonstrated graph-based topic clustering is in its several parameters of SR-MCL which are set to default values in the experiments. Changing of these parameter values could affect the clustering results.

For topic extraction, WordNet and IDF are used to extract semantically-similar keywords from each cluster to represent a topic. The results are visualized using graphs to investigate the relationships between reviews sharing the same topics and between keywords co-occurring within the same and different topics. Highly overlapped topics are detected by bunches of edges within the graphs.

7 Conclusion

In this paper, a topic clustering approach based on SR-MCL is applied to cluster user reviews on tourist attractions. Using BCubed metrics to determine the fit between the resulting clusters and those of the gold standard dataset, we found that the approach is competitive to LDA in terms of clustering user reviews.

While LDA explicitly requires the number of clusters as prior knowledge, SR-MCL relies on the Inflate parameter (r) to dynamically determine the number of clusters. The higher value of r results in more clusters being generated. We set r to 1.5 as in [1] to demonstrate the robustness of the parameter against different text corpora. This value of r results in relatively high precision and recall.

For topic extraction, WordNet and IDF are utilized to extract keywords from each cluster which represents a topic. To explore the resulting topics and investigate their semantic relationships, we illustrate document-topic and term-topic relationships as graphs. The term-topic relationship, moreover, allows the consumers to possibly catch the main themes within the attractions.

Our future work focuses on semantic improvement and exploration on the effects of other SR-MCL parameters on the dataset.

Acknowledgments. This research project was supported by Faculty of Information and Communication Technology, Mahidol University. The study was carried out under the research framework of Mahidol University.

References

1. Aker, A., et al.: A graph-based approach to topic clustering for online comments to news. In: Ferro, N., et al. (eds.) ECIR 2016. LNCS, vol. 9626, pp. 15–29. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30671-1_2
2. Alghamdi, R., Alfalqi, K.: A survey of topic modeling in text mining. Int. J. Adv. Comput. Sci. Appl. **6** (2015). https://doi.org/10.14569/IJACSA.2015.060121

3. Alkhodair, S.A., Fung, B.C.M., Rahman, O., Hung, P.C.K.: Improving interpretations of topic modeling in microblogs. J. Assoc. Inf. Sci. Technol. **69**(4), 528–540 (2018). https://doi.org/10.1002/asi.23980
4. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Inf. Retrieval **12**(4), 461–486 (2009). https://doi.org/10.1007/s10791-008-9066-8
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003). http://dl.acm.org/citation.cfm?id=944919.944937
6. DeMaere, M.Z., Darling, A.E.: Deconvoluting simulated metagenomes: the performance of hard- and soft- clustering algorithms applied to metagenomic chromosome conformation capture (3C). PeerJ **4**, e2676 (2016). https://doi.org/10.7717/peerj.2676
7. van Dongen, S.M.: Graph clustering by flow simulation. Ph.D. thesis, University of Utrecht, The Netherlands (2000). https://dspace.library.uu.nl/handle/1874/848
8. Dorow, B., Widdows, D.: Discovering corpus-specific word senses. In: 10th Conference of the European Chapter of the Association for Computational Linguistics (2003). http://aclweb.org/anthology/E03-1020
9. Grant, C.E., George, C.P., Kanjilal, V., Nirkhiwale, S., Wilson, J.N., Wang, D.Z.: A topic-based search, visualization, and exploration system. In: FLAIRS Conference (2015)
10. Griffiths, T.L., Steyvers, M.: Finding scientific topics. In: Proceedings of the National Academy of Sciences, vol. 101, pp. 5228–5235. National Academy of Sciences (2004). https://doi.org/10.1073/pnas.0307752101
11. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. In: Proceedings of the 11th Eurographics/IEEE - VGTC Conference on Visualization, EuroVis 2009, pp. 983–998. The Eurographics Association and Wiley, Chichester (2009). https://doi.org/10.1111/j.1467-8659.2009.01450.x
12. Ji, Z., Pi, H., Wei, W., Xiong, B., Woźniak, M., Damasevicius, R.: Recommendation based on review texts and social communities. A hybrid model. IEEE Access **7**, 40416–40427 (2019). https://doi.org/10.1109/ACCESS.2019.2897586
13. Jindal, V.: A personalized Markov clustering and deep learning approach for Arabic text categorization. In: Proceedings of the ACL 2016 Student Research Workshop, pp. 145–151. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/P16-3022
14. Jurgens, D., Klapaftis, I.: SemEval-2013 task 13: word sense induction for graded and non-graded senses. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp. 290–299. Association for Computational Linguistics (2013). http://aclweb.org/anthology/S13-2049
15. Litvin, S., Hoffman, L.M.: Responses to consumer-generated media in the hospitality marketplace: an empirical study. J. Vacation Mark. **18**, 135–145 (2012). https://doi.org/10.1177/1356766712443467
16. Llewellyn, C., Grover, C., Oberlander, J.: Improving topic model clustering of newspaper comments for summarisation. In: Proceedings of the ACL 2016 Student Research Workshop, pp. 43–50. Association for Computational Linguistics, Berlin, August 2016. http://anthology.aclweb.org/P/P16/P16-3007
17. Phuong, D.V., Phuong, T.M.: A keyword-topic model for contextual advertising. In: Proceedings of the Third Symposium on Information and Communication Technology, SoICT 2012, pp. 63–70 (2012). https://doi.org/10.1145/2350716.2350728

18. Satuluri, V., Parthasarathy, S.: Scalable graph clustering using stochastic flows: applications to community discovery. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 737–746. ACM, New York (2009). https://doi.org/10.1145/1557019.1557101
19. Satuluri, V., Parthasarathy, S., Ucar, D.: Markov clustering of protein interaction networks with improved balance and scalability. In: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology, BCB 2010, pp. 247–256. ACM, New York (2010). https://doi.org/10.1145/1854776.1854812
20. Shih, Y.K., Parthasarathy, S.: Identifying functional modules in interaction networks through overlapping Markov clustering. Bioinformatics **28**(18), i473–i479 (2012). https://doi.org/10.1093/bioinformatics/bts370