exceed $250K$ keys, allowing us to store FacetCache in main memory for faster retrievals.

Concept Tips. To illustrate the particular relevance of a blog post to a search query, each result item is accompanied by concept tips – a subset of its key concepts that are semantically related to concepts in the query. Concept tips are essentially an intersection between facets and keywords:

$$\bigcup_{c_i \in \text{Query}} \text{FacetCache}[c_i].keys \bigcap \text{KeyConcepts(post)}.keys$$

Facet Views. Two kinds of presentational views are supported for facets: a list view and a clustered view. In *list view*, facets are displayed in a flat list, ranked by their relevance to a search query. In *clustered view* facets are grouped into named topical clusters according to their pair-wise semantic similarity. Clustered view is especially illustrative for a large set of facets. This allows the user to quickly observe general categories related to a search query and then go for finer-grained facets of a certain category.

Clustered view is shown in Figure 1 on the right pane. Internally, the view is represented as a map from a general concept that names a topical cluster to weighted facets that constitute the cluster. Algorithm 2 illustrates the computation of the clustered view, accepting as input weighted facets for a search query: $\text{ClusteredView}(\cup_{c_i \in \text{Query}} \text{FacetCache}[c_i])$. Topical clusters are computed from a semantic graph induced by facets using the Girvan-Newman algorithm [2]. For each cluster that is semantically dense enough to exceed a certain threshold λ, a general concept is inferred for naming the cluster using the Spreading Activation algorithm [8]. The remaining clusters are merged under an administrative concept "Misc". The threshold λ is chosen experimentally to balance between making clusters that exceed it contain reasonably related facets and keeping the "Misc" cluster moderate in size.

Algorithm 2 ClusteredView

Input: WeightedConcepts
Output: View

```
1: View = ∅
2: graph = SemanticGraph(WeightedConcepts.keys)
3: ClustersSet = GirvanNewman(graph)
4: for cluster ∈ ClustersSet do
5:    density = 1/|cluster|^2 Σ_{c_i,c_j ∈ cluster} sim(c_i, c_j)
6:    if (density > λ) then
7:       View[SpreadingActivation(cluster)] = cluster
8:    else
9:       View["Misc"] = View["Misc"] ⋃ cluster
```

What a Blog is About. To give the user an illustrative impression on a subject of a blog at a glance, we display its primary topics inferred from its content and key concepts for each topic. This data is computed with the same Algorithm 2, only invoked for weighted key concepts of the blog. Unlike conventional tag clouds, the clustered view is more structured and involves background knowledge on semantics.

4.3 Recommendation

Blognoon offers recommendations with respect to both individual blog posts and whole blogs, internally computed via concept search. When the user opens a blog post, a search query is constructed from key concepts of the post and search results are presented as relevant recommendations. In the same way, recommendations for blogs are performed using key concepts of a currently opened blog. Recommendations offered by Blognoon constitute a valuable tool for topic exploration, for they are based on non-trivial semantic relations between concepts and allow the user discover relevant material which is difficult to locate otherwise.

5. RELATED WORK

Kosmix [7] aims at providing topic exploration functionality in scope of the whole Web, by combining deep web crawl and federated search. Alternatively, Blognoon addresses only blog data, which makes the crawling approach sufficient in our case and allows us to employ sophisticated preprocessing of crawled data for providing *concept* search which includes finding information that is semantically similar to a query, not only textually similar. For semantically ambiguous words in data sources, Kosmix achieves their sense disambiguation implicitly, by either relying on particular data sources or by adding a particular meaning to a query as another word. Blognoon performs explicit word sense disambiguation for all words in data [9] as a preprocessing step.

Generation of a topic page for a query is investigated in [1], primarily for the biographical domain. Their approach to detecting multiple aspect of a topic based on term clustering is somewhat reminiscent to clustered view of key concepts in Blognoon. We however do not aim at generating a single page that would cover a whole topic, but instead offer semantic-driven navigational features to facilitate topic exploration.

6. REFERENCES

[1] N. Balasubramanian and S. Cucerzan. Topic pages: An alternative to the ten blue links. In *Proc. IEEE Int. Conf. on Semantic Computing, 2010*.

[2] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6):066111, Dec 2004.

[3] E. Gabrilovich and S. Markovitch. Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Int. Res.*, 34:443–498, March 2009.

[4] M. Grineva, M. Grinev, and D. Lizorkin. Extracting key terms from noisy and multitheme documents. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 661–670, New York, NY, USA, 2009. ACM.

[5] M. A. Hearst, M. Hurst, and S. T. Dumais. What should blog search look like? In *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*, pages 95–98, New York, NY, USA, 2008. ACM.

[6] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

[7] A. Rajaraman. Kosmix: high-performance topic exploration using the deep web. *Proc. VLDB Endow.*, 2(2):1524–1529, 2009.

[8] Z. Syed, T. Finin, and A. Joshi. Wikipedia as an ontology for describing documents. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March 2008.

[9] D. Turdakov and P. Velikhov. Semantic relatedness metric for Wikipedia concepts based on link analysis and its application to word sense disambiguation. In *SYRCoDIS*, 2008.