Fast Concept Mention Grouping for Concept Map–based Multi-Document Summarization

Tobias Falke and **Iryna Gurevych**
Research Training Group AIPHES and UKP Lab
Department of Computer Science, Technische Universität Darmstadt
https://www.aiphes.tu-darmstadt.de

Abstract

Concept map–based multi-document summarization has recently been proposed as a variant of the traditional summarization task with graph-structured summaries. As shown by previous work, the grouping of coreferent concept mentions across documents is a crucial subtask of it. However, while the current state-of-the-art method suggested a new grouping method that was shown to improve the summary quality, its use of pairwise comparisons leads to polynomial runtime complexity that prohibits the application to large document collections. In this paper, we propose two alternative grouping techniques based on locality sensitive hashing, approximate nearest neighbor search and a fast clustering algorithm. They exhibit linear and log-linear runtime complexity, making them much more scalable. We report experimental results that confirm the improved runtime behavior while also showing that the quality of the summary concept maps remains comparable.[1]

1 Introduction

Concept maps are labeled graphs with nodes representing concepts and edges showing relationships between them (Novak and Gowin, 1984). Following earlier work on the automatic extraction of concept maps from text (Rajaraman and Tan, 2002; Valerio and Leake, 2006; Villalon, 2012; Zubrinic et al., 2015), concept maps have recently been promoted as an alternative representation for summaries (Falke and Gurevych, 2017; Handler and O'Connor, 2018). In the corresponding task, concept map–based multi-document summarization (CM-MDS), a set of documents has to be automatically summarized as a concept map that does not exceed a pre-defined size limit.

An important subtask of CM-MDS is *concept mention grouping*, in which all mentions that refer to a specific concept should be grouped together. Without grouping, duplicates can appear in a summary concept map that make the map harder to understand and that waste valuable space.

To approach the mention grouping subtask, Falke et al. (2017) proposed to make pairwise coreference classifications between mentions and to induce a partitioning from those predictions. Their experiments showed that this leads to better summary concept maps, establishing the current state-of-the-art for CM-MDS. However, the computational costs of the approach are high, as it exhibits a $\mathcal{O}(n^4)$ worst-case time complexity. When the number of documents that should be summarized is large, applying that technique can quickly become impractical. But exactly for those large document sets, a summary would be most helpful.

As the first contribution of this paper, we propose two faster grouping techniques. First, we apply *locality sensitive hashing (LSH)* (Charikar, 2002) to word embeddings in order to find similar mentions without making all pairwise comparisons. That directly leads to a simple $\mathcal{O}(n)$ grouping method. Second, we also propose a novel grouping technique that combines the hashing approach with a fast partitioning algorithm called *Chinese Whispers (CW)* (Biemann, 2006). It has $\mathcal{O}(n \log n)$ time complexity and the advantage of being more transparently controllable.

Since the reduced complexity of the two proposed techniques is gained through approximations, the resulting grouping could of course be of lower quality. As the second contribution of this paper, we therefore carry out end-to-end experiments in the context of CM-MDS to analyze this trade-off. We compare both techniques against the state-of-the-art approach in automatic and manual evaluations. For both, we observe orders of mag-

[1] Code used for experiments available at https://github.com/UKPLab/naacl2019-cmaps-lshcw

Proceedings of NAACL-HLT 2019, pages 695–700
Minneapolis, Minnesota, June 2 - June 7, 2019.

nitude faster runtimes with only small reductions in summary quality. In the future, the techniques could also be applied beyond CM-MDS to speed up other similarity-based partitioning problems in NLP and its applications.

2 Problem and Reference Approach

Given a set of concept mentions M identified in the input documents, the goal of concept mention grouping is to derive a partitioning C of M such that for every set of mentions in C, the set contains all mentions and only mentions of one unique concept. Let n denote the number of mentions $|M|$.

Previous work on concept map mining used stemming (Villalon, 2012), substring matches (Valerio and Leake, 2006) or WordNet (Aguiar et al., 2016) to detect coreferences between mentions. Falke et al. (2017) combined several of those features, including semantic similarities based on WordNet (Miller et al., 1990), latent semantic analysis (Deerwester et al., 1990) and word2vec embeddings (Mikolov et al., 2013), in a log-linear classifier to predict coreferences of mentions.

Since such pairwise predictions can be inconsistent, e.g. the model might classify (m_1, m_2) and (m_2, m_3) as coreferent, but not (m_1, m_3), Falke et al. (2017) further induce a transitive relation from the predictions to obtain a valid partitioning of M. They note that simply ignoring conflicting negative classifications by building the transitive closure over all positive ones typically yields undesired partitionings in which too many mentions are being lumped together. Following previous work on related NLP tasks (Barzilay and Lapata, 2006; Denis and Baldridge, 2007), they instead formulate an integer linear program (ILP) to find the transitive relation that maximally agrees with all pairwise predictions. However, as the resulting ILPs cannot be efficiently solved on the data they work with, they propose a local search algorithm that incrementally improves a greedy solution rather than finding the optimal partitioning,

This technique requires making classifications for all pairs of mentions in $\mathcal{O}(n^2)$ time and running the local search, which has a worst-case complexity of $\mathcal{O}(n^4)$. As we will show in Section 6, that can quickly become prohibitively expensive.

3 Locality Sensitive Hashing

The central idea of LSH is that specific families of hash functions can approximately preserve similarities. Charikar (2002) introduced such a family for cosine similarity between vectors.

3.1 Approximating Cosine Similarity

Let u, v be k-dimensional vectors. First, choose d unit random vectors $r_1, \ldots, r_d$ of k dimensions by sampling every dimension independently from a standard normal distribution. Then, for a vector u, compute a d-dimensional bit vector $h(u)$, the hash, with the i-th dimension defined as

$$h(u)_{[i]} = \begin{cases} 1 & : u \cdot r_i \geq 0 \\ 0 & : u \cdot r_i < 0, \end{cases} \quad (1)$$

where $u \cdot r_i$ is the dot product with the i-th random vector. The Hamming distance ham between two hashes $h(u)$ and $h(v)$, i.e. the number of differing bits, can then be used to approximate the cosine similarity of u and v (Charikar, 2002):

$$\frac{u \cdot v}{|u||v|} \approx \cos\left(\frac{\mathrm{ham}(h(u), h(v))}{d}\, \pi\right) \quad (2)$$

The longer the hashes are, i.e. the larger d is, the more accurate is the estimation of the similarity.

In the past, LSH has been successfully used to speed up a range of NLP tasks, including noun similarity list construction (Ravichandran et al., 2005), word sense induction (Mouton et al., 2009), gender classification (van Durme, 2012) and text classification (Bollegala et al., 2018).

3.2 Naive Partitioning

Given the mapping h from vectors to their bit hashes, we can partition a set of vectors by hash identity. Every unique hash becomes a group consisting of all vectors mapped to that hash. Since the hashes reflect similarity, the most similar vectors will be grouped together. The parameter d controls the degree of grouping: the smaller it is, the less unique hashes and thus fewer groups exist.

In order to apply this technique to concept mention grouping, every mention $m \in M$ has to be represented by a vector in a space where the cosine similarity is indicative of coreference. Since the classifier of Falke et al. (2017) already uses cosine similarity of word2vec embeddings as a feature, we also use those vectors for LSH.[2] Both the computation of the hashes and building groups can be done with a single pass over the mentions. Assuming d and k to be fixed, the overall time complexity of the grouping technique is thus $\mathcal{O}(n)$.

[2]Following their work, we represent a mention by the mean of the embedding vectors of the mention's tokens.

4 Fast Nearest Neighbor Partitioning

When grouping similar elements together, one typically wants to control the degree of grouping by defining a similarity threshold δ. For the naive LSH-based partitioning, we can only set d, which does not directly correspond to a similarity. Therefore, we propose a second, more transparent grouping technique with this property.

4.1 Approximate Nearest Neighbor Search

Given vectors and their LSH-based hashes, we can use *approximate nearest neighbor search (ANNS)* to find pairs with a cosine similarity of at least δ (Charikar, 2002; Ravichandran et al., 2005) without making all pairwise comparisons:

1. Sample q permutations of the bit hashes.
2. For each permutation, sort all mentions M according to their permuted hashes.
3. In each sorted list, estimate the cosine similarity of each $m \in M$ with the next b mentions based on the hashes. Keep pairs with a similarity of at least δ.

Since comparing neighbors in a sorted list of bit hashes will primarily find those that differ in the last positions, the random permutations are the key part of the algorithm that ensures similar hashes differing at varying positions are found. Rather than comparing each vector to all others in $\mathcal{O}(n^2)$, only qb comparisons are made for each. The dominant part becomes the sort, resulting in $\mathcal{O}(n \log n)$ time complexity as q and b are constants.

4.2 Chinese Whispers Partitioning

Using ANNS we can obtain an undirected graph of mentions connected with edges if their similarity is at least δ. However, as Falke et al. (2017) observed, simply taking the transitive closure over these pairs tends to yield too big groups that lump many mentions of different concepts together.

Rather than relying on the expensive $\mathcal{O}(n^4)$ local search of Falke et al. (2017) to address this problem, we here resort to the fast graph partitioning algorithm CW (Biemann, 2006). Given a graph $G = (V, E)$, it proceeds as follows:

1. Label nodes initially as $l(v_i) = i \;\; \forall \, v_i \in V$.
2. Iterate over V in randomized order. For each $v \in V$, set $l(v)$ to the label most frequent among the nodes reachable via a direct edge.
3. If at least one label changed, repeat step 2.

While it cannot be guaranteed in general, the algorithm typically converges to a stable labeling after a few iterations. Then, nodes having the same label form a group of the partitioning. In contrast to the local search, CW does not directly optimize the objective function proposed by Falke et al. (2017), however, we empirically found that it yields partitionings that score very well with regard to that objective. To guarantee termination, the number of iterations is bound by a parameter ϵ. Then, CW iterates at most ϵ times over n nodes and their at most $n - 1$ edges, resulting in $\mathcal{O}(n^2)$ complexity.

4.3 Combination

For concept mention grouping, we combine these techniques as follows: First, we represent each mention with a vector and compute its LSH-based hash. Second, we use ANNS to find pairs with a similarity of at least δ. Finally, we partition the resulting nearest neighbor graph with CW.

That grouping technique has four parameters δ, d, q and b. While δ determines the degree of grouping, d influences the quality of the similarity estimates and q and b define the size of the search space explored to find nearest neighbors. Note that the construction of the nearest neighbor graph guarantees that a node has at most qb edges, reducing the runtime of CW to $\mathcal{O}(n)$ in this setting. The runtime behavior of the combination is therefore dominated by ANNS and thus $\mathcal{O}(n \log n)$.

5 Experimental Setup

We evaluate the proposed concept mention grouping techniques for the task of CM-MDS.

Data and Metrics We use the benchmark corpus introduced by Falke and Gurevych (2017), the only existing dataset with manually created reference summary concept maps. It provides reference summaries for document sets of web pages on 30 different topics. As metrics, we compute the ROUGE and METEOR variants proposed with the dataset and also perform a human evaluation following the protocol of Falke et al. (2017).

Implementation As the *reference*, we use the state-of-the-art pipeline of Falke et al. (2017).[3] We test the naive LSH-based partitioning (*LSH-only*)

[3] https://github.com/UKPLab/ijcnlp2017-cmaps

	Average		Smallest		Largest	
Approach	Count	Runtime	Count	Runtime	Count	Runtime
Mentions	*5299*		*2475*		*13572*	
Reference	4029	3h 12m 32s	1847	24m 21s	10131	22h 48m 08s
LSH-only	3694	1s	1752	1s	7827	2s
LSH-CW	4085	23s	1875	11s	9861	58s

Table 1: Concept mention grouping runtimes on average and for the smallest and largest set. *Count* is the number of concepts after grouping the mentions given in the first row. Runtimes are measured on the same machine.

	METEOR			ROUGE-2		
Approach	Pr	Re	F1	Pr	Re	F1
Reference	15.1	17.3	16.1	9.4	11.9	10.4
lemma-only	13.9	15.4	*14.6*	8.2	8.6	*8.3*
w2v-only	14.1	16.2	*15.0*	8.3	9.9	*8.9*
LSH-only	14.9	16.9	15.8	9.1	11.2	9.9
LSH-CW	14.9	17.1	15.9	8.2	10.9	9.3

Table 2: Evaluation results for summary concept maps. Italics denote F1-scores that are significantly different from *Reference* (exact permutation test, $\alpha = 0.05$).

Comparison	Fo	Gr	Me	NR
Reference vs. LSH-only	47.3	47.3	46.7	42.7
Reference vs. LSH-CW	57.3	58.0	58.0	56.7
LSH-CW vs. LSH-only	52.7	50.7	50.0	47.3

Table 3: Human summary preferences, shown as the percentage of annotators preferring the first option.

and the combined approach (*LSH-CW*) by substituting them into that pipeline. For a fair comparison, we use the same 300-dimensional word2vec embeddings (Mikolov et al., 2013) for LSH that have also been used in the log-linear model.

Tuning In the reference pipeline, the regularization constant of the scoring SVM was tuned with leave-one-out cross-validation on the training set. For LSH-only, we use the same procedure to tune d (together with regularization) and found $d = 17$ to be best (testing 10, 11, ..., 25). For LSH-CW, where four hyper-parameters have to be set, running cross-validation for the whole grid is too expensive. We instead evaluate a grid of 130 d/q/b/δ-combinations by concept F1-score after grouping and tune the SVM with cross-validation only for the three best settings, leading to the parameters $d = 200, q = 20, b = 200, \delta = .89$.

6 Results

Runtime Table 1 shows the runtimes for grouping concept mentions.[4] It demonstrates two problems of the reference: First, even on the smallest document set (37 docs, 50k tokens), the grouping already takes hours. And second, on the biggest set (42 docs, 220k tokens), the runtime grows to almost a day, illustrating the analyzed time complexity. Applying the technique to more documents quickly becomes infeasible. Our newly proposed techniques, LSH-only and LSH-CW, are orders of magnitude faster in absolute terms and also show a more moderate runtime growth as expected given their preferable time complexity.

Quality A crucial question is which price we have to pay for improving runtimes through approximations. Table 2 shows the automatic evaluation results for the created summaries. We included *lemma-only*, a baseline from previous work using lemmatization for grouping, and *w2v-only*, a variation of the reference grouping approach that uses embeddings as the only feature in the coreference classifier. The latter is important for comparison, as it uses the same information as the LSH-based techniques. While lemma-only and w2v-only perform significantly worse than the reference, the two LSH-based techniques come much closer to the more expensive reference.

Table 3 shows the results of our human evaluation. Following previous work, we collected pairwise preferences among the created summaries via Mechanical Turk (150 per pairing) for the dimensions focus (Fo), grammaticality (Gr), meaningfulness (Me) and non-redundancy (NR).[5] As shown, the preferences we collected are almost balanced and annotators repeatedly noted during the study that the summaries are very similar. None of the 12 preferences are significant at $\alpha =$

[4] Measured on an Intel Xeon ES-2620 2.1GHz processor.

[5] We payed $0.60 per comparison and anonymized worker IDs. The study was approved by the university's ethics committee and we obtained informed consent from participants.

0.05 (binomial test), showing that the alternative summary concept maps are practically indistinguishable. In contrast, Falke et al. (2017) observed preferences of up to 79% in their study.

Conclusion Based on the automatic and human evaluations, we conclude that both fast grouping techniques proposed in this paper do not substantially decrease the quality of the summaries. Since there is also no clear difference between LSH-only and LSH-CW, we recommend both techniques, which allows practitioners to choose between more transparency or even faster runtimes.

Future Work The comparison of w2v-only and the reference in Table 2 reveals that relying only on word2vec and dropping the other features of the log-linear model hurts performance, suggesting that also adding the remaining features to the LSH techniques could lead to further improvements. However, all other features of the reference model are pairwise features, which makes it difficult to incorporate them in the LSH-based techniques that only use mention features. As an alternative direction, one could instead rely on more powerful word embeddings. While we used word2vec to ensure comparability to previous work, using more recent embedding methods such as fastText (Bojanowski et al., 2017), InferSent (Conneau et al., 2017) or ELMO (Peters et al., 2018) seems to be worth exploring in the future.

7 Summary

In this paper, we proposed two fast concept mention grouping techniques for CM-MDS, the direct application of LSH and a novel combination of LSH and Chinese Whispers. Our analysis and experiments show that they are orders of magnitude faster than previous techniques with only small effects the quality of the resulting summary concept maps. Using these techniques, summary concept maps can now be created for much larger document sets than what was possible before.

Acknowledgements

We would like to thank Kevin Mayer for his support during preliminary experiments leading to this paper. This work has been supported by the German Research Foundation as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1.

References

Camila Z. Aguiar, Davidson Cury, and Amal Zouaq. 2016. Automatic Constrution of Concept Maps from Texts. In *Proceedings of the 7th International Conference on Concept Mapping*, pages 20–30, Tallinn, Estonia.

Regina Barzilay and Mirella Lapata. 2006. Aggregation via Set Partitioning for Natural Language Generation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 359–366, New York, NY, USA.

Chris Biemann. 2006. Chinese Whispers - An Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems. In *Proceedings of TextGraphs: The First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, New York, NY, USA.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Danushka Bollegala, Vincent Atanasov, Takanori Maehara, and Ken-Ichi Kawarabayashi. 2018. ClassiNet - Predicting Missing Features for Short-Text Classification. *ACM Transactions on Knowledge Discovery from Data*, 12(5):1–29.

Moses S. Charikar. 2002. Similarity Estimation Techniques From Rounding Algorithms. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, pages 380–388, Montréal, Canada.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Pascal Denis and Jason Baldridge. 2007. Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–243, Rochester, NY, USA.

Tobias Falke and Iryna Gurevych. 2017. Bringing Structure into Summaries: Crowdsourcing a Benchmark Corpus of Concept Maps. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2951–2961, Copenhagen, Denmark.

Tobias Falke, Christian M. Meyer, and Iryna Gurevych. 2017. Concept-Map-Based Multi-Document Summarization using Concept Coreference Resolution and Global Importance Optimization. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pages 801–811, Taipei, Taiwan.

Abram Handler and Brendan O'Connor. 2018. Relational Summarization for Corpus Analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1760–1769, New Orleans, LA, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, NV, USA.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An On-line Lexical Database*. *International Journal of Lexicography*, 3(4):235–244.

Claire Mouton, Guillaume Pitel, Gaël de Chalendar, and Anne Vilnat. 2009. Unsupervised Word Sense Induction from Multiple Semantic Spaces with Locality Sensitive Hashing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 287–291, Borovets, Bulgaria.

Joseph D. Novak and D. Bob Gowin. 1984. *Learning How to Learn*. Cambridge University Press, Cambridge, MA, USA.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, New Orleans, LA, USA.

Kanagasabai Rajaraman and Ah-Hwee Tan. 2002. Knowledge Discovery from Texts: A Concept Frame Graph Approach. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 669–671, McLean, VA, USA.

Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Function for High Speed Noun Clustering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 622–629, Ann Arbor, MI, USA.

Alejandro Valerio and David B. Leake. 2006. Jump-Starting Concept Map Construction with Knowledge Extracted from Documents. In *Concept Maps: Theory, Methodology, Technology. Proceedings of the 2nd International Conference on Concept Mapping*, pages 296–303, San José, Costa Rica.

Benjamin van Durme. 2012. Streaming Analysis of Discourse Participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 48–58, Jeju Island, Korea.

Jorge J. Villalon. 2012. *Automated Generation of Concept Maps to Support Writing*. Ph.D. Thesis, University of Sydney.

Krunoslav Zubrinic, Ines Obradovic, and Tomo Sjekavica. 2015. Implementation of method for generating concept map from unstructured text in the Croatian language. In *23rd International Conference on Software, Telecommunications and Computer Networks*, pages 220–223, Split, Croatia.