Hierarchical topic model inference by community discovery on word co-occurrence networks

Eric Austin[1,2][0000−0002−7532−9455], Amine Trabelsi[3][0000−0002−1852−5265], Christine Largeron[4][0000−0003−1059−4095], and Osmar R. Zaïane[1,2][0000−0002−0060−5988]

[1] University of Alberta, Edmonton AB T6G 2R3, Canada
{eaustin, zaiane}@ualberta.ca
[2] Alberta Machine Intelligence Institute, Edmonton AB T5J 3B1, Canada
[3] Université de Sherbrooke, Sherbrooke QC, J1K 2R1, Canada
Amine.Trabelsi@USherbrooke.ca
[4] Université Jean Monnet, Saint-Etienne, France
largeron@univ-st-etienne.fr

Abstract. The most popular topic modelling algorithm, Latent Dirichlet Allocation, produces a simple set of topics. However, topics naturally exist in a hierarchy with larger, more general super-topics and smaller, more specific sub-topics. We develop a novel topic modelling algorithm, Community Topic, that mines communities from word co-occurrence networks to produce topics. The fractal structure of networks provides a natural topic hierarchy where sub-topics can be found by iteratively mining the sub-graph formed by a single topic. Similarly, super-topics can by found by mining the network of topic hyper-nodes. We compare the topic hierarchies discovered by Community Topic to those produced by two probabilistic graphical topic models and find that Community Topic uncovers a topic hierarchy with a more coherent structure and a tighter relationship between parent and child topics. Community Topic is able to find this hierarchy more quickly and allows for on-demand sub- and super-topic discovery, facilitating corpus exploration by researchers.

Keywords: Topic Modelling · Information Networks · Graphs · Natural Language Processing · Data Mining.

1 Introduction

Topic modelling discovers the themes of collections of unstructured text documents. Topics can act as features for document classification and indices for information retrieval. However, one of the most important functions of these topics is to assist in the exploration and understanding of large corpora. Researchers in all fields and domains seek to better understand the main ideas and themes of document collections too large for a human to read and summarize. This requires topics that are interpretable and coherent to human users.

Interpretability is a necessary but not sufficient condition for a good topic model. Topics naturally exist in a hierarchy. There are larger, more general super-topics and smaller, more specific sub-topics. “Sports” is a valid topic in that it represents a concept. “Football” and “the Olympics” are also topics. They are not completely distinct from “sports” but rather are sub-topics that fall within sports, i.e. they are child topics of the “Sports” parent topic in the topic hierarchy. Topics also relate to each other to varying degrees. The “movie” topic is more similar to the “television” topic than the “food” topic. This relationship structure is also key to understanding the topical content of a corpus. Topic modelling methods that simply provide the user with a set of topics are not as useful and informative as those that can provide this hierarchy and structure.

Recently, a new domain has emerged where topics can provide utility: conversational agents, which are computer programs that can carry on a human-level conversation. The conversation is an end in itself; the purpose of speaking with a conversational agent is to converse, to be entertained, to express emotion and be supported. The awareness and use of the topics of discussion are key abilities that an agent must possess to be able to carry on a conversation with a human. Previous work has used the detected topic of conversation to enrich the a conversational agent’s responses [12]. However, more can be done with topics to improve the abilities of a conversational agent given the right topic model that provides a topic hierarchy and structure. It can be used to detect and control topic drift in the conversation so that the agent’s responses make sense in context. If the user is engaged with the current topic, then the agent can stay on topic or detect sub-topics to focus the conversation. The agent can detect super-topics to broaden the range of conversation. The agent should be able to move to related topics or, if the user becomes bored or displeased, jump to dissimilar topics. This type of control over the flow of the conversation is crucial to human communication and is needed for human-computer interaction as well.

The most widely used topic model, Latent Dirichlet Allocation (LDA), only provides a simple set of topics without a hierarchy or structure and has other drawbacks. The number of topics must be specified, requiring multiple runs with different numbers of topics to find the best topics. It performs poorly on short documents. Different runs on the same corpus can produce different topics, especially if the order of the documents is different [25]. Common terms can appear in many different topics, reducing the uniqueness of topics [31].

Neural networks have pushed forward the state-of-the-art in topic modelling. While neural topic models have produced topics of greater coherence, they retain many of the weaknesses of LDA, such as the need to specify the number of topics, while having a tendency to find models with many redundant topics [7] and demanding greater computational resources and specialized hardware.

These drawbacks have inspired us to search for a new approach to topic modelling. We desire a method that can operate quickly on commodity hardware and that provides not only a set of topics but their relationships and a hierarchical structure. It is natural to take an information network-based approach given the growing importance of relational data and graphs in representing complex systems [34]. Our topic modelling algorithm, Community Topic (CT), mines communities from networks constructed from term co-occurrences. These topics are collections of vocabulary terms and are thus interpretable by humans. The fractal nature of the network representation provides a natural topic hierarchy and structure. The topic hyper-vertices form a network with connections of varying strength between the topic vertices derived from the aggregated edges between their constituent word vertices. Super-topics can be mined from this topic network. Each topic itself is also a sub-graph with regions of varying density of connections. This sub-graph can be mined to find sub-topics. Our algorithm has only a single hyperparameter and can run quickly on simple hardware which makes it ideal for researchers from all fields for exploring a document collection.

In this paper, we review related work on topic modelling. We describe our algorithm, how it constructs term co-occurrence networks, and how it mines topics from these networks. We describe how it discovers the topic hierarchy and how this can be done on-the-fly as needed by the user. We empirically evaluate our algorithm and compare it to two probabilistic graphical topic models. Our results show that our approach is able to find a topic hierarchy with a more coherent structure and a tighter relationship between parent and child topics. Community Topic is able to find this hierarchy more quickly and allows for on-demand sub- and super-topic discovery.

2 Related Work

Topic modelling emerged from the field of information retrieval and research to reduce the dimensionality of and more effectively represent documents for indexing, query matching, and document classification. The performance of topic models on these tasks has been surpassed by deep neural models but topic models have become extremely popular tools of applied research both inside and outside of computing science [18]. One early approach is Latent Semantic Analysis (LSA) [10] which decomposes the term-by-document matrix to find vectors representing the latent semantic structure of the corpus and can be viewed as (uninterpretable) topics that relate terms and documents. Another matrix decomposition method is Non-negative Matrix Factorization [23]. Researchers unsatisfied with the lack of a solid statistical foundation to LSA developed Probabilistic Latent Semantic Analysis (pLSA) [17] which posits a generative probabilistic model of the data with the topics as the latent variables.

A drawback of pLSA is that the topic mixture is estimated separately for each document. Latent Dirichlet Allocation (LDA) [5], not to be confused with Linear Discriminant Analysis, was developed to remedy this. LDA is a fully generative model as it places a Dirichlet prior on the latent topic mixture of a document. The probability of a topic z given a document d, $p(z|d;\theta)$, is a multinomial distribution over the topics parameterized by θ where θ is itself a random variable sampled from the prior Dirichlet distribution. The number of topics must be specified and the model provides no topic hierarchy or structure.

There have been many methods developed that attempt to improve upon LDA. Promoting named entities to become the most frequent terms in the document has been tried [22]. In [39], the authors use a process to identify and re-weight words that are topic-indiscriminate. To improve the performance of

LDA on tweets, the authors of [27] pool tweets into longer documents. The MetaLDA model [41] incorporates meta information such as document labels. The author-topic model [37] extends LDA by conditioning the topic mixture on document author. The Correlated Topic Model (CTM) [3] models the correlations between topics. The Dynamic Topic Model [4] allows for the modelling of topic evolution over time. Most relevant to our work are two methods that discover a hierarchy of topics. The Hierarchical LDA model (HLDA) [16] models the topic hierarchy using a tree structure. The depth of the tree must be specified but the number of topics is discovered. A flexible generalization of LDA is the Pachinko Allocation Model (PAM) [24]. Like HLDA, PAM allows for a hierachy of topics but this hierarchy is represented by a directed acyclic graph rather than a tree of fixed depth, allowing for a variety of relationships between topics and terms in the hierarchy, although this structure must be specified by the user.

In recent years, new topic models have emerged based on neural networks. The Embedded Topic Model (ETM) [11] combines word embeddings trained using the Skip-gram algorithm [29] with the LDA probabilistic generative model. Another approach is to use a variational autoencoder (VAE) [20][21] to learn the probability distributions of a generative probabilistic model, as with the neural variational document model (NVDM) [28], the stick-breaking variational autoencoder (SB-VAE) [30], ProdLDA [36], and Dirichlet-VAE [7]. These models discover topics that are qualitatively different than those found by traditional LDA, although there is debate as to whether they are truly superior [18]. Neural models that provide a topic hierarchy have also been developed. In [40], the authors develop Weibull hybrid autoencoding inference (WHAI) to model multiple layers of priors for deep LDA and thus multiple layers in a topic hierarchy. The number of hyperparameters, complicated training process, and need for special hardware makes this type of model unsuitable for applied researchers seeking a tool for corpus exploration.

3 Community Topic

We call our community detection-based topic modelling algorithm Community Topic (CT). The design of CT was driven by the results of experimentation detailed in our previous paper [2] and the version presented here is the final result of this process of experimentation. The algorithm has three main steps.

3.1 Co-occurrence network construction

First, a network is constructed from the document corpus with terms as vertices. An edge exists between a pair of vertices v_i and v_j if the terms t_i and t_j co-occur in the same sentence. The weights of edges are derived from the frequency of co-occurrence. One method is to use the raw count as the edge weight. However, this does not adjust for the frequency of the terms themselves so more common terms will tend to have higher edge weights. An alternative weighting scheme is to use normalized pointwise mutual information (NPMI) between terms (Eq. 1).

$$NPMI(t_i, t_j) = \frac{log\frac{p(t_i,t_j)}{p(t_i)p(t_j)}}{-log(p(t_i, t_j))} \quad (1)$$

NPMI assigns higher values to pairs of terms t_i and t_j whose co-occurrence, $p(t_i, t_j)$, is more frequent than what would be expected if their occurrences in the texts were random, $p(t_i)p(t_j)$. This is normalized to adjust for the frequencies of the terms in the corpus. The edges of the network are thresholded at 0, i.e. those edges with weights ≤ 0 are removed from the network. This is because the community mining algorithm we will use to discover topics uses modularity Q [32] to discover the more densely connected regions of the network. This formula uses the product of the weighted degrees of two vertices to determine the expected value of the strength of their connection if the graph were random, which does not work if a vertex has a negative weighted degree.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{i,j} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j) \quad (2)$$

Here m is the sum of weights of all edges in the network, $A_{i,j}$ is the weight of the edge connecting v_i and v_j, k_i (k_j) is the sum of weights of edges incident to v_i (v_j), C_i (C_j) is the assigned community of v_i (v_j), and δ is an indicator function that returns 1 when the two arguments are equal and 0 otherwise.

The distribution of edge weights differs greatly between the raw count and NPMI. The raw count weights follow a power law distribution with the vast majority of edges having very low weight and very few edges with very high weight. This mirrors the power law distribution of term frequencies. Given this distribution of term frequencies, a given edge weight value can carry very different information. An edge weight of 2 could indicate a significant relationship between two terms that occur 5 times each. Between two terms that occur hundreds of times each, an edge weigh of 2 would be noise. When we convert the edge weights to NPMI values, they are scaled to the range [-1,+1] and high values are assigned to edges that represent frequent co-occurrence relative to the frequencies of the connected terms. This distribution resembles a bell curve. We see very few edge weights ≤ 0 that will be removed by thresholding. This indicates that conditioned on co-occurring at least once, two terms are likely to co-occur more often than would be expected by chance. In our experiments we found slightly better results using the NPMI edge weights. We refer the interested reader to our previous work [2] for a visualization of these edge weight distributions.

3.2 Community Mining

Once the co-occurrence network is constructed, CT discovers topics by applying a community detection algorithm. A community is a group of vertices that have a greater density of connections among themselves than they do to vertices outside the group. Many community detection algoritms exist and have been surveryed in other work [9][14][15]. CT employs the Leiden algorithm [38] as this was found to work best in experimentation. The Leiden algorithm has a resolution parameter that is used to set the scale at which communities are discovered. Smaller values of this parameter lead to larger communities being found and larger values lead to smaller communities. This represents the only hyperparameter necessary for CT and is less a value that needs to be carefully tuned for good performance but is rather a way for the user to get communities of a desired size.

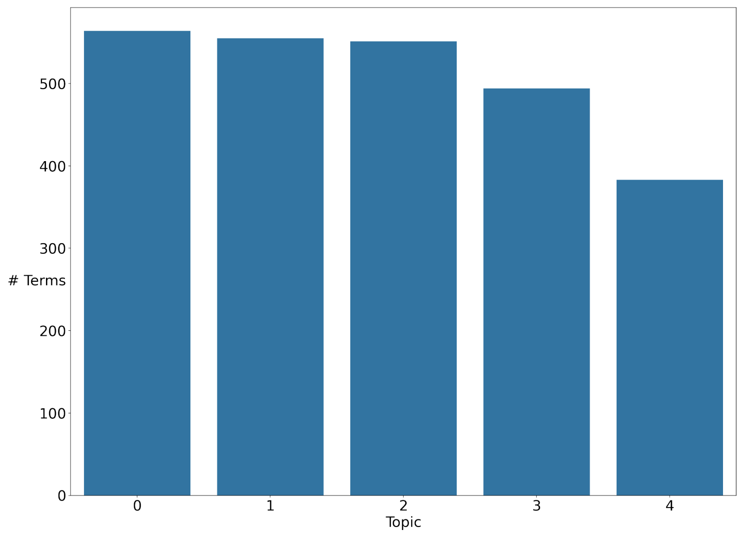

Fig. 1. Distribution of community sizes found by Leiden with resolution parameter 1.0.

Figure 1 Shows the distribution of community sizes found when using a Leiden resolution parameter of 1.0 on the BBC News dataset[5]. CT returns 5 large topics that correspond to the five article categories of the dataset. In Figure 2, we see the a resolution parameter of 1.5 returns a greater number of small topics with a greater varience of topic size, from hundreds of terms to just a few.

3.3 Topic Filtering and Term Ordering

Once the communities are discovered, small communities of size 2 or less are removed as outliers. Probabilistic graphical topic models such as LDA produce topics that are probability distributions over vocabulary terms. The most important terms for a topic are simply those that have the highest probabilities. The communities discovered by the Leiden algorithm are sets of vertices, so CT needs a way of ranking the terms represented by those vertices. To do so, we take advantage of the graph representation and use internal weighted degree to rank vertices/terms, which is calculated as the sum of weights of edges incident to a

[5] https://www.kaggle.com/competitions/learn-ai-bbc/data

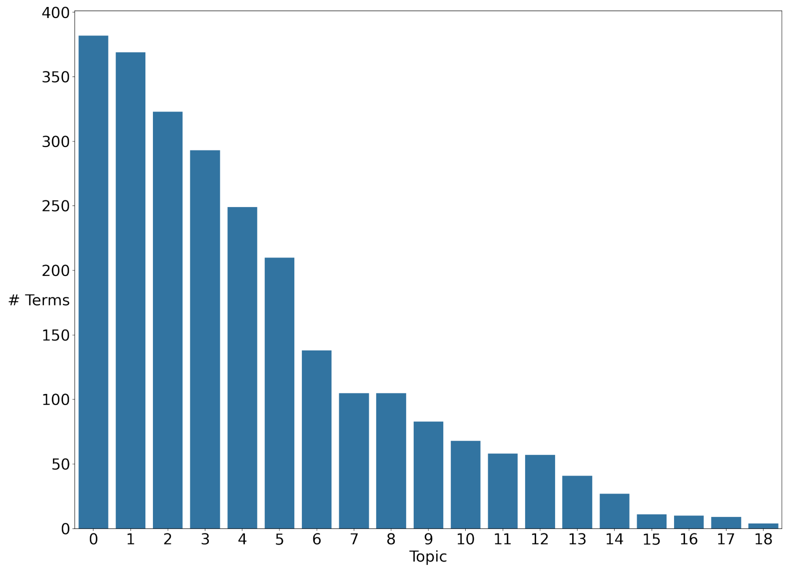

Fig. 2. Distribution of community sizes found by Leiden with resolution parameter 1.5.

vertex that connect to another vertex in the same community/topic. This gives higher values to terms that connect strongly to many terms in the same topic and are thus most representative of that topic. Once the filtering and ordering is complete, the set of topics is returned to the user.

3.4 Topic Hierarchy

This basic formulation of CT produces a set of topics like vanilla LDA. However, there exists a natural structure to the graph representation and it is straightforward to adapt CT to return a hierarchy. By iteratively applying community detection to each topic sub-graph, CT discovers the next level of the topic hierarchy. This can be done to a specified depth or we can allow CT to uncover the entire hierarchy by stopping the growth of the topic tree once the produced sub-topics are smaller than three terms. An example of 3 levels of topics discovered on the BBC corpus is show in Figure 3. The level 1 topics correspond to the 5 article categories of the corpus. Level 2 and then 3 show increasingly specific sub-topics.

The topic hierarchy can also be constructed in a bottom-up fashion. If a low Leiden resolution parameter is initially used, CT produces many small topics. Applying community detection to the network of topic vertices groups these small sub-topics into super-topics. We can see an example of this in Figure 4 shows the clustering of the initial small topics discovered on the BBC corpus into super-topics which roughly correspond to the 5 article categories of the corpus.

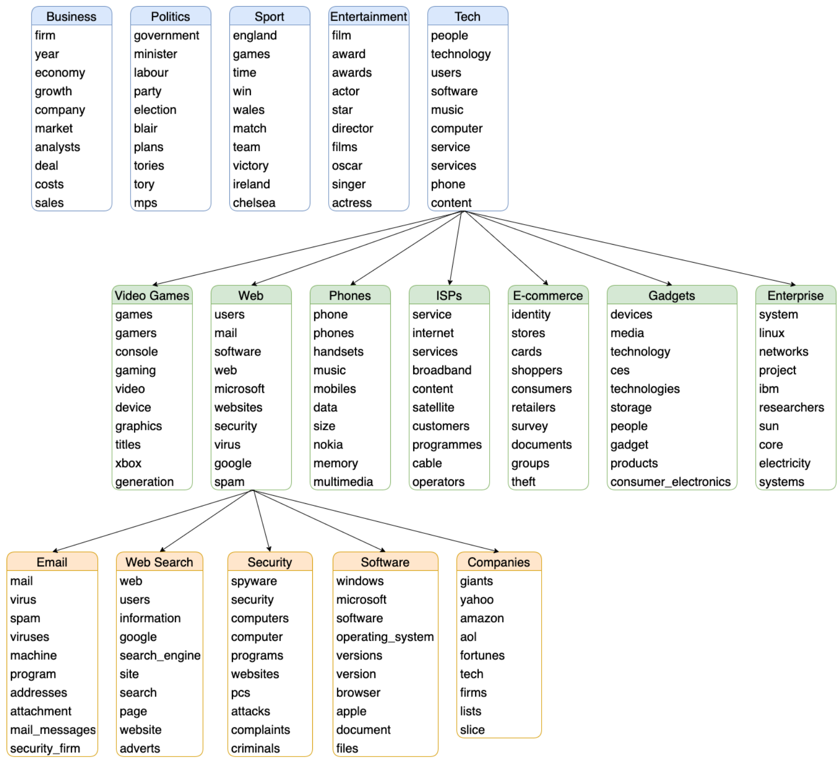

Fig. 3. Hierarchy of BBC corpus topics found by iteratively applying CT algorithm.

4 Empirical Evaluation

In this section we compare CT to two probabilistic graphical topic models, HLDA and PAM[6]. As the implementation of PAM only allows for two non-root topic layers in the hierarchy we generate a three-level hierarchy for each algorithm for fair comparison, where level 0 is the root topic of all terms in the corpus, level 1 are the super-topics, and level 2 are the sub-topics. PAM requires the number of super- and sub-topics to be specified. We used the number of topics discovered by CT at each level for PAM.

4.1 Datasets

We use three datasets to evaluate the different topic modelling approaches: 20Newsgroups[7], Reuters-21578[8], and BBC News[9]. The 20Newsgroups dataset

[6] https://bab2min.github.io/tomotopy/v0.12.2/en/
[7] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html
[8] https://huggingface.co/datasets/reuters21578
[9] https://www.kaggle.com/competitions/learn-ai-bbc/data

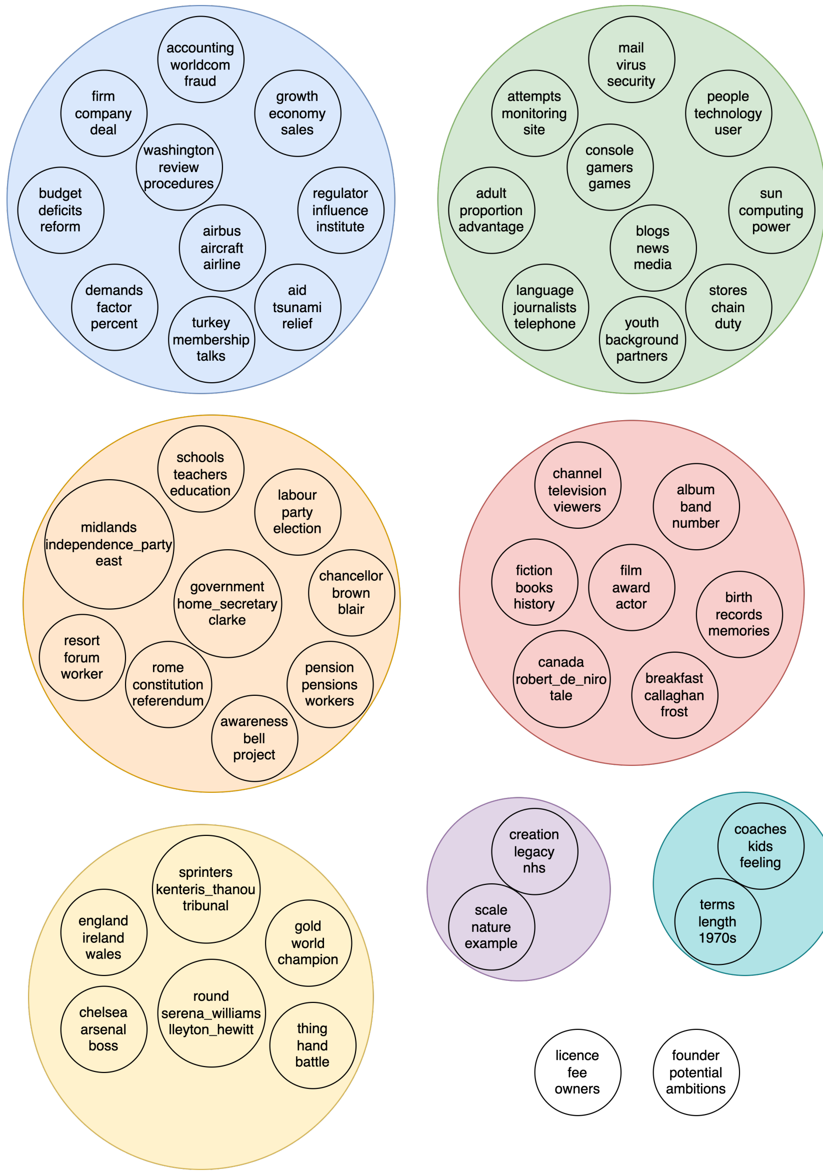

Fig. 4. Super-topics found by applying community detection on network of small topics.

consists of 18,846 posts on the Usenet discussion platform which come from 20 different topics such as "atheism" and "hockey". The Reuters-21578 dataset consists of 21,578 financial articles published on the Reuters newswire in 1987 and have economic and financial topics such as "grain" and "copper". The BBC News dataset consists of 2225 articles in five categories: “business”, “entertainment”, “politics”, “sport”, and “tech”.

4.2 Preprocessing

We use spaCy[10] to lowercase and tokenize the documents and to identify sentences, parts-of-speech (POS), and named entities. We only detect noun-type entities which are merged into single tokens e.g. the terms “united”, “states”, “of”, and “america” become “united_states_of_america”. While stemming and lemmatization have been commonly used in the topic modelling literature, the authors of [35] found that they do not improve topic quality and hurt model stability so we do not stem or lemmatize. We remove stopwords and terms that occur in $> 90\%$ of documents. Following [18], we remove terms that appear in fewer than $2(0.02|d|)^{1/log10}$ documents. It was shown in [26] that topic models constructed from noun-only corpora were more coherent so we detect and tag parts-of-speech to be able to filter out non-noun terms as in [8]. This is intuitive as adjectives and verbs can be used in many different contexts, e.g. one can “play the piano”, “play baseball”, “play the stock market”, and “play with someone’s heart”, but music, sports, finance, and romance are separate topics. Even with nouns there are issues with polysemy, i.e. words with multiple meanings and thus multiple different common contexts. To help with this problem, we use Gensim[11] to extract meaningful n-grams [6]. An n-gram is a combination of n adjacent tokens into a single token so that a term such as “microsoft_windows” can be found and the computer operating system can be distinguished from the windows of a building. We apply two iterations so that longer n-grams such as “law_enforcement_agencies” can be found.

4.3 Evaluation Metrics

To measure the quality of the topics produced by each model, we use two coherence measures: C_V [33] and C_{NPMI} [1]. Both measures have been shown to correlate with human judgements of topic quality with C_V having the strongest correlation [33]. Even though C_V has stronger correlation that C_{NPMI} with human evaluations, C_{NPMI} is more commonly used in the literature [18], possibly due to the extra computation required by C_V. We prefer the C_V measures as, in addition to being more highly correlated with human judgement, it considers the similarity of the contexts of the terms, not just their own co-occurrence. We use Gensim[12] to compute both measures. Each dataset has a train/test split. We train all models on the train documents and evaluate using the test documents. We use the standard 110-term window for C_V and 10-term window for C_{NPMI}. We use the top 5 terms of each topic for evaluation

[10] https://spacy.io/
[11] https://radimrehurek.com/gensim/
[12] https://radimrehurek.com/gensim/models/coherencemodel.html

To measure the quality of the topic hierarchy, we use two measures proposed in [19]: topic specialization and hierarchical affinity. Topic specialization measures the distance of a topic's probability distribution over terms from the general probability distribution of all terms in the corpus given by their occurrence frequency. We expect topics at higher levels in the hierarchy closer to the root to be more general and less specialized and topics further down the hierarchy to be more specialized. Hierarchical affinity measures the similarity between a super-topic and a set of sub-topics. We expect higher affinity between a parent topic and its children and lower affinity between a parent topic and sub-topics which are not its children.

HLDA produces topics at both levels that are probability distributions over vocabulary terms and are thus compatible with our evaluation metrics without modification. CT produces a list of terms ranked by the internal weighted degree. To calculate specialization and affinity, we convert these to probability distributions by dividing each value by the sum of the values. The super-topics discovered by PAM are distributions over sub-topics. We convert these to distributions over terms by taking the expectation for each term in the sub-topics given the super-topic distribution over sub-topics. Each PAM super-topic distribution gives some non-zero probability to all sub-topics so we need a way to distinguish children from non-children. We do this by taking the top 6 most likely sub-topics as the children of a super-topic since we are positing a topic hierarchy with an average of 6 sub-topics per super-topic.

5 Results

Using a Leiden resolution parameter of 1.0, CT finds 5 or 6 super-topics on all datasets and 5, 6, or 7 sub-topics per super topic and we use these average values to guide the PAM model. HLDA finds hundreds of super-topics and about 3 times as many sub-topics. This tendency to find many small topics at all levels leads to poor performance on our evaluation metrics and leads to a poor hierarchy where it is common for a child topic to appear in more documents than its parent. PAM performs better, but benefits from using the number of topics discovered by CT.

CT is the fastest of the algorithms, finding the topic hierarchy in under 5 seconds on all datasets. HLDA takes between 30 seconds and 5 minutes while PAM ranges from 10 seconds to 2 minutes. All experiments were run on the same laptop with 2.7 GHz dual core processor and 8 GB RAM.

The coherence results are presented in Table 1. We can see that CT achieves the highest coherence scores on all datasets as measured by both metrics except for C_{NPMI} on the 20Newsgroup corpus where PAM comes out on top. PAM acheives the second highest scores in all other cases. HLDA is a distant third with much lower scores. This demonstrates that the topics found by CT will be more interpretable to a human user.

Figure 5 shows the specialization scores for each algorithm on the three datasets. We see that both the super-topics (level 1) and the sub-topics (level 2)

Table 1. Coherence scores for CT, HLDA, PAM on three document corpora. Bold indicates best score for each metric and dataset.

	BBC		20Newsgroups		Reuters	
	C_V	C_{NPMI}	C_V	C_{NPMI}	C_V	C_{NPMI}
CT	**0.641**	**0.079**	**0.645**	0.044	**0.702**	**0.182**
HLDA	0.448	-0.162	0.444	-0.133	0.451	-0.093
PAM	0.600	0.063	0.636	**0.090**	0.555	0.056

found by HLDA have a very high specialization. This is consistent with the large number of topics found at both levels but does not match our intuition that topics higher in the hierarchy should be general. PAM produces general topics at level 1 and more specialized topics at level 2, however the super-topics are so general and similar to the overall frequency distribution as to not provide useful information for the user. CT also produces sub-topics that are more specialized than the super-topics. Unlike PAM, the super-topics are themselves specialized and thus useful and informative themselves.

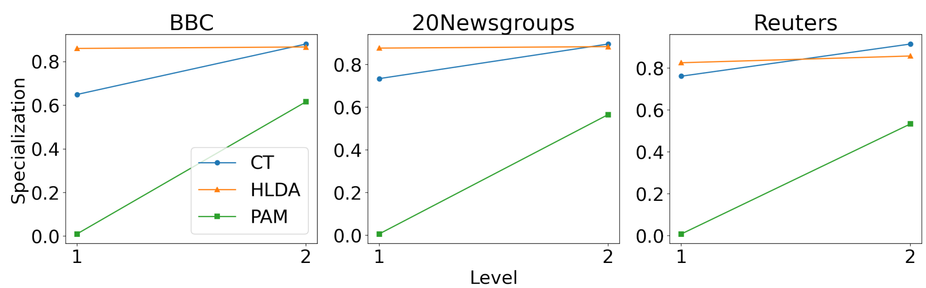

Fig. 5. Topic specialization scores for CT, HLDA, and PAM on three corpora.

Figure 6 shows the hierarchical affinity scores for each algorithm on the three datasets. We see that HLDA has a higher affinity between parent topics and their children than non-children. However, the affinity is very low so the relationship between a super-topic and its sub-topics is very weak. PAM has the opposite problem with high affinities between parent topics and both child and non-child topics. This is because PAM super-topics are distributions over all sub-topics and is consistent with the super-topics being non-specialized. CT parent topics exhibit a high affinity with their children and zero affinity with non-children. This is because the sub-topics are a partition of the super-topic and thus do not overlap with any other super-topic.

Our experimental results show that CT produces the most coherent and thus interpretable topics and the best topic hierarchy. CT topic hierarchies exhibit higher specialization for sub-topics than super-topics but with enough specialization at both levels to make the topics useful. CT super-topics have a high affinity

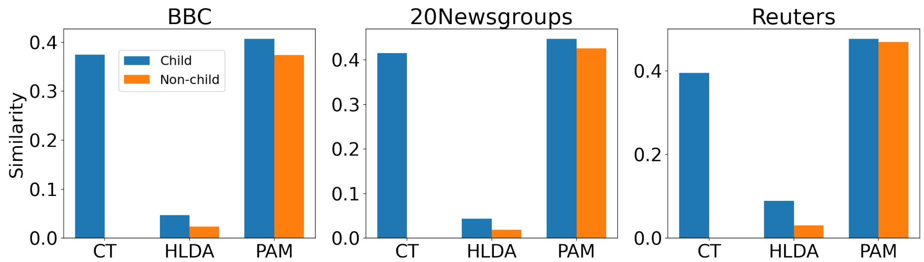

Fig. 6. Hierarchical affinity scores between parent and children and between parent and non-children for CT, HLDA, and PAM on three corpora.

with their own sub-topics and no affinity with non-child sub-topics. CT is able to produce this coherent topic structure in less time than the other algorithms on commodity hardware.

6 Conclusion

We have presented our novel hierarchical topic modelling algorithm, CT. This method is based on community mining of word co-occurrence networks and is thus fast and takes advantage of the natural network structure. Our experiments show that CT produces more coherent topics and a more cohesive topic hierarchy than either HLDA or PAM. The features of CT make it an ideal tool of corpus exploration and to guide the conversation of a chat bot.

In future, we would like to extend CT by allowing for overlapping topics. Currently topics are partitions of the vocabulary. A method such as the persona splitting of [13] that creates multiple instances of a vertex would allow for terms to fall into multiple topics. While our method has shown good performance on automated metrics, the real test of a topic model is in its utility for downstream tasks and we plan to integrate CT into a conversational agent to demonstrate its utility.

References

1. Aletras, N., Stevenson, M.: Evaluating topic coherence using distributional semantics. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers. pp. 13–22 (2013)
2. Austin, E., Zaïane, O., Largeron, C.: Community topic: Topic model inference by consecutive word community discovery. In: Proceedings of COLING 2022, the 32nd International Conference on Computational Linguistics (2022)
3. Blei, D., Lafferty, J.: Correlated topic models. Advances in Neural Information Processing Systems **18**, 147 (2006)
4. Blei, D., Lafferty, J.: Dynamic topic models. In: Proceeding of the 23rd International Conference on Machine Learning. pp. 113–120 (2006). https://doi.org/10.1145/1143844.1143859

5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3**(Jan), 993–1022 (2003). https://doi.org/10.1016/B978-0-12-411519-4.00006-9
6. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. Proceedings of GSCL **30**, 31–40 (2009)
7. Burkhardt, S., Kramer, S.: Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. Journal of Machine Learning Research **20**(131), 1–27 (2019)
8. Chen, J., Zaïane, O.R., Goebel, R.: An unsupervised approach to cluster web search results based on word sense communities. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. vol. 1, pp. 725–729. IEEE (2008). https://doi.org/10.1109/WIIAT.2008.24
9. Coscia, M., Giannotti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining: The ASA Data Science Journal **4**(5), 512–546 (2011). https://doi.org/10.1002/sam.10133
10. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41**(6), 391–407 (1990). https://doi.org/10.1002/(sici)1097-4571(199009)41:6¡391::aid-asi1¿3.0.co;2-9
11. Dieng, A.B., Ruiz, F.J.R., Blei, D.M.: Topic Modeling in Embedding Spaces. Transactions of the Association for Computational Linguistics **8**, 439–453 (07 2020). https://doi.org/10.1162/tacl_a_00325
12. Dziri, N., Kamalloo, E., Mathewson, K., Zaïane, O.R.: Augmenting neural response generation with context-aware topical attention. In: Proceedings of the First Workshop on NLP for Conversational AI. pp. 18–31 (2019). https://doi.org/10.18653/v1/W19-4103
13. Epasto, A., Lattanzi, S., Paes Leme, R.: Ego-splitting framework: From non-overlapping to overlapping clusters. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 145–154 (2017)
14. Fortunato, S.: Community detection in graphs. Physics Reports **486**(3-5), 75–174 (2010). https://doi.org/10.1016/j.physrep.2009.11.002
15. Fortunato, S., Hric, D.: Community detection in networks: A user guide. Physics Reports **659**, 1–44 (2016). https://doi.org/10.1016/j.physrep.2016.09.002
16. Griffiths, T., Jordan, M., Tenenbaum, J., Blei, D.: Hierarchical topic models and the nested chinese restaurant process. Advances in Neural Information Processing Systems **16** (2003)
17. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 50–57 (1999). https://doi.org/10.1145/312624.312649
18. Hoyle, A., Goel, P., Hian-Cheong, A., Peskov, D., Boyd-Graber, J., Resnik, P.: Is automated topic model evaluation broken? the incoherence of coherence. Advances in Neural Information Processing Systems **34** (2021)
19. Kim, J.H., Kim, D., Kim, S., Oh, A.: Modeling topic hierarchies with the recursive chinese restaurant process. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. pp. 783–792 (2012)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the International Conference on Learning Representations (ICLR) (2014)
21. Kingma, D.P., Welling, M., et al.: An introduction to variational autoencoders. Foundations and Trends in Machine Learning **12**(4), 307–392 (2019). https://doi.org/10.1561/9781680836233

22. Krasnashchok, K., Jouili, S.: Improving topic quality by promoting named entities in topic modeling. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 247–253 (2018). https://doi.org/10.18653/v1/P18-2040
23. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999). https://doi.org/10.1038/44565
24. Li, W., McCallum, A.: Pachinko allocation: Dag-structured mixture models of topic correlations. In: Proceedings of the 23rd International Conference on Machine Learning. p. 577–584. ICML '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1143844.1143917
25. Mantyla, M.V., Claes, M., Farooq, U.: Measuring lda topic stability from clusters of replicated runs. In: Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 1–4 (2018). https://doi.org/10.1145/3239235.3267435
26. Martin, F., Johnson, M.: More efficient topic modelling through a noun only approach. In: Proceedings of the Australasian Language Technology Association Workshop 2015. pp. 111–115 (2015)
27. Mehrotra, R., Sanner, S., Buntine, W., Xie, L.: Improving lda topic models for microblogs via tweet pooling and automatic labeling. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 889–892 (2013). https://doi.org/10.1145/2484028.2484166
28. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. In: International Conference on Machine Learning. pp. 1727–1736. PMLR (2016)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems **26** (2013)
30. Nalisnick, E., Smyth, P.: Stick-breaking variational autoencoders. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
31. Nan, F., Ding, R., Nallapati, R., Xiang, B.: Topic modeling with wasserstein autoencoders. arXiv preprint arXiv:1907.12374 (2019). https://doi.org/10.18653/v1/P19-1640
32. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69**(2), 026113 (2004). https://doi.org/10.1103/physreve.69.026113
33. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. pp. 399–408 (2015). https://doi.org/10.1145/2684822.2685324
34. Sakr, S., Bonifati, A., Voigt, H., Iosup, A., Ammar, K., Angles, R., Aref, W., Arenas, M., Besta, M., Boncz, P.A., et al.: The future is big graphs: a community view on graph processing systems. Communications of the ACM **64**(9), 62–71 (2021). https://doi.org/10.1145/3434642
35. Schofield, A., Mimno, D.: Comparing apples to apple: The effects of stemmers on topic models. Transactions of the Association for Computational Linguistics **4**, 287–300 (2016). https://doi.org/10.1162/tacl_a_00099
36. Srivastava, A., Sutton, C.: Autoencoding variational inference for topic models. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
37. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 306–315 (2004). https://doi.org/10.1145/1014052.1014087

38. Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific Reports **9**(1), 1–12 (2019). https://doi.org/10.1038/s41598-019-41695-z
39. Yang, K., Cai, Y., Chen, Z., Leung, H.f., Lau, R.: Exploring topic discriminating power of words in latent dirichlet allocation. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2238–2247 (2016)
40. Zhang, H., Chen, B., Guo, D., Zhou, M.: WHAI: weibull hybrid autoencoding inference for deep topic modeling. In: 6th International Conference on Learning Representations (ICLR) (2018)
41. Zhao, H., Du, L., Buntine, W., Liu, G.: Metalda: A topic model that efficiently incorporates meta information. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 635–644 (2017). https://doi.org/10.1109/ICDM.2017.73