Automated Concept Map Extraction from Text

Anonymous EMNLP submission

Abstract

Concept maps are summaries of nodes and relations from text in a directed graph format that can foster students' learning and understanding. However, manually constructing them is a challenging task. Automatic concept map extraction methods have emerged, standardly with a pipeline approach consisting of methods to extract entities and their relations. Yet, existing methods face efficiency limitations: 1) they are not capable of dealing with big corpora, 2) they are not open-access architectures, 3) they rely on the existence of annotated datasets. To bridge these gaps, we introduce a novel, modularized and open-source methods for concept map extraction that addresses efficiency by using semantic and sub-symbolic techniques with a new preliminary summarisation component. Moreover, we compare the pipeline approaches with three end-to-end Large Language Models methods. The best models for our pipeline and our end-to-end baseline achieve state-of-the-art results on METEOR metrics, with F1 scores of 25.69 and 28.5 respectively and on ROUGE-2 recall, with scores of 24.26 and 24.3. This contribution advances the task of automated concept map extraction, opening doors to wider applications supporting learning. The code is open-access and available[1].

1 Introduction

A concept map is a visual representation that displays directed relations between different concepts in a graph, as shown in Figure 1. Concept maps facilitate the integration of new information with preexisting knowledge (Canas et al., 2001), promote active processing of information (Novak, 1990), enhance long-term memory retention and foster better understanding and critical thinking (Novak and Gowin, 1984). These multiple functionalities make them valuable not only for educational purposes but also in clinical settings for addressing and rehabilitating language disorders. Notably, creating concept maps is among the most effective strategies for assisting children with language disorders (Dexter and Hughes, 2011; Ausubel et al., 1968; Nesbit and Adesope, 2006). Additionally, its potential applications extend beyond learning, as demonstrated by several studies in information retrieval and knowledge representation (Villalon, 2012; Leake, 2006).

The manual creation of concept maps from text is challenging and impractical due to the time-consuming nature of the task. As a result, there has recently been significant attention given to the automatic extraction of concept maps from text (de Aguiar et al., 2016; Falke et al., 2016; Falke and Gurevych, 2017; Falke et al., 2017; Falke, 2019). However, the automatic construction of con-

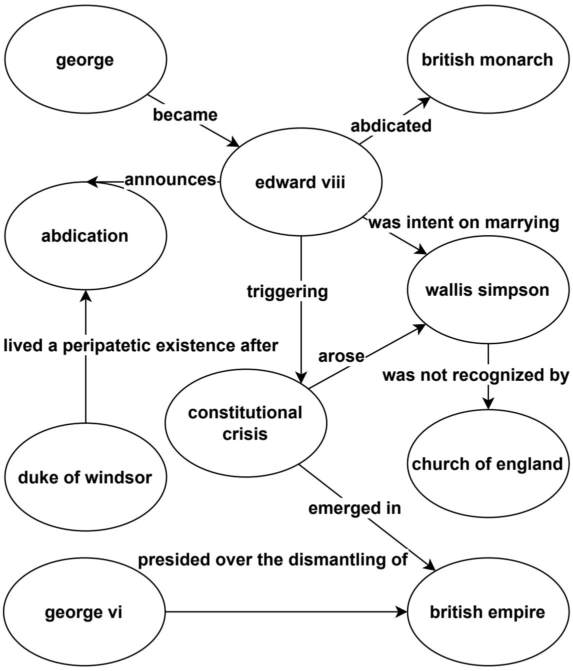


Figure 1: An example of a concept map, showing the concepts and relations between them. It was created based on the folder 320 of the WIKI dataset (Falke, 2019).

[1] https://github.com/vs1rr/automatic_concept_map_extraction/tree/master

cept maps, traditionally implemented as a pipeline including components such as concept and relation extraction, grouping and labeling and importance ranking, still exhibit several shortcomings. First, current methods struggle with efficiency, being often limited to processing small document collections and unable to handle large-scale datasets. Second, they depend on the availability of annotated datasets to implement supervised models. Third, their code or architecture is not openly accessible.

Inspired by past and current approaches, we maintain a pipeline architecture, and introduce an open access, cost-efficient and modularized system, based on semantic techniques and Large Language Models (LLMs). To the best of our knowledge, we provide two main contributions and novelties: 1) We integrate sub-symbolic techniques, such as neural-based relation extraction, for automated concept map extraction; 2) We introduce a preliminary summarization and importance ranking components to reduce the search space. Furthermore, we address the need for efficiency which currently relies on heavily annotated corpora, by fine-tuning a sequence-to-sequence model based on BART for the relation extraction part. Lastly, we compare our model and results to three end-to-end LLMs baselines. Our code and experiments are openly accessible[2]. We achieve state-of-the-art results on the METEOR metrics and ROUGE-2 Recall.

2 Related Work

Concept map extraction can be framed as a summarization task where the summary is in a graph format. In this work, we focus on concept map extraction from unstructured data. The literature conventionally portrays the automatic extraction of concept maps from text as a multi-step process, involving sub-tasks such as concept and relation extraction, and sub-graph selection. Existing works can be divided into two types of methods: the ones with multiple documents as inputs, namely the Concept Map - Multi Document Summarization (CM-MDS), and the ones with a single document as input, namely the Concept Map - Document Summarization (CM-DS) (Falke et al., 2017).

Early research efforts focused on CM-DS (Oliveira et al., 2001). This approach laid the groundwork by not only extracting relations between concepts from a text file, but also by extrapolating rules about the knowledge at hand. Subsequent studies like Leake (2006) employed unsupervised methods with deep syntactic parsing for concept selection. These methods primarily used term frequencies to assign a document to the most probable concept map among a set of options, enhancing the accuracy of concept selection. Kowata et al. (2010) further focused on extracting concept maps from Portuguese news articles. This work pioneered the use a comprehensive pipeline approach that included text segmentation, tokenization, part-of-speech tagging, core elements candidate recognition, dependency interpretation, and concept map construction. de Aguiar et al. (2016) introduced a sophisticated pipeline approach that integrated grammar rules, co-reference resolution, and concept ranking based on occurrence frequency. Lastly, Bayrak and Dal (2024) introduced a new heuristic approach to extract concept maps from Turkish texts.

For CM-MDS, Rajaraman and Tan (2002) pioneered the field by utilizing regular expressions and term-frequency-based grouping to construct a concept-map-based knowledge base from text documents. They used Named Entity Recognition, extracted noun-verb-noun triples using a POS tagger and handcrafted rules, disambiguated them with WordNet, and clustered them. Their approach was integrated into a system and validated through experimental studies. Zouaq et al. (2011) later defined specific patterns over dependency syntax representations to enhance entity extraction. Their work highlighted the usefulness of concept map mining in ontology learning. Žubrinić et al. (2015) extended the CM-MDS task by introducing a heuristic approach for summarizing concept maps from legal documents written in Croatian. This was a significant advancement that demonstrated the adaptability of CM-MDS techniques to other languages and domain-specific document types.

Lastly, Falke et al. (2017; 2017; 2019) made significant contributions to the field and their datasets serve as the main benchmark for the CM-MDS task. Their model leverages predicate-argument structures and automatic models for German and English, achieving state-of-the-art performance. Their pipeline approach including five distinct steps: (1) concept and relation extraction, relying on Open Information Extraction; (2) Concept Mention Grouping and Labeling with greedy search optimization (3) Relation Mention Grouping,

[2] https://github.com/vs1rr/automatic_concept_map_extraction/tree/master

Labeling and Selection using lemmatization (4) Importance Estimation with a Ranking Support Vector Machine (5) Concept Map Construction using Integer Linear Programming.

Authors	Task	Language	Method	SE	IR	EE	RE
Oliveira et al. (2001)	S	EN	L				✓
Rajaraman and Tan (2002)	M	EN	L		✓	✓	
Leake (2006)	S	EN	LS			✓	
Kowata et al. (2010)	S	PR	LS				
Zouaq et al. (2011)	M	EN	L		✓	✓	✓
Zubrinic et al. (2012)	M	CR	LS	post		✓	✓
Qasim et al. (2013)	M	EN	LS			✓	✓
Žubrinić et al. (2015)	M	CR	LS		✓	✓	
de Aguiar et al. (2016)	S	EN	LS	post		✓	✓
Falke (2019)	M	EN,DE	LS	post	✓	✓	✓
Nugumanova et al. (2021)	M	EN,KK,RU	L			✓	✓
Bayrak and Dal (2024)	M	TR	LS		✓	✓	✓
Our pipeline approach	M,S	EN	LS	pre	✓	✓	✓

Table 1: Comparison of existing pipeline methods for CM-DS (S) and CM-MDS (M) tasks from text data to our pipeline. For the header: SE: Summary Extraction, IR: Importance Ranking, EE: Entity Extraction, RE: Relation Extraction. For the Language: EN: English, DE: German, KK: Kazakh, RU: Russian, CR: Croatian, PR: Portuguese. For the method: linguistic tools (L), linguistic and statistical tools (LS). For Summary Extraction (SE): pre: SE occurs before entity and relation extraction, while $post$: SE occurs after.

Table 1 summarizes existing methods for both CM-DS and CM-MDS. These works showcase the evolution from basic term frequency methods to more complex pipelines. However, existing approaches rely on symbolic or machine learning methods, lacking the incorporation of advanced neural techniques that can enhance relation extraction accuracy. Additionally, no previous studies have introduced the preliminary summarization and importance ranking components that we use to reduce the search space by focusing on the most important content. Furthermore, we fine-tune a sequence-to-sequence models for the relation extraction sub-task, which can address the challenges posed by the need for heavily annotated corpora and improve efficiency. Lastly, we provide a comprehensive comparison between our method and end-to-end LLM-based methods. By understanding and building upon existing methodologies, we introduce an open access, cost-efficient and modularized system, based on symbolic and sub-symbolic techniques.

3 Methods

We present two methods for automated concept map extraction from text. Following the literature, we first propose a pipeline-based approach. We introduce a modular, open-access method with four components, three optional and one mandatory : (1) Summary Extraction, (2) Importance Ranking, (3) Entity Extraction and (4) Relation Extraction. (1), (2) and (3) can be deactivated in the pipeline, while (4) is always required, as depicted in Figure 2. While (3) and (4) were standardly used in previous approaches, we are the first ones to propose (1) and (2) as primary steps for more efficiency. Second, we introduce three end-to-end LLMs baselines.

3.1 Pipeline approach

While the state-of-the-art method (Falke, 2019) used as last step graph summarization, we investigate whether adding an (1) importance ranking and (2) summarization steps at the very beginning of our pipeline can yield better results and alleviate the search space.

Pre-processing. This step transforms all text to lowercase, removes punctuation, and filters out noise-prone information such as web links.

Summary Extraction. We integrate methods for extractive and abstractive summarization. Extractive summarization extracts key sentences from the original text, while abstractive summarization generates a concise summary using new phrases and sentences. For extractive summarization, we use LexRank (Erkan and Radev, 2004)[3]. We choose this method as it was previously used for concept-based extractive summarization (Chitrakala et al., 2018) and it leverages graph-based and ranking methods particularly relevant for our task. For abstractive summarization, we use *gpt-3.5-turbo-0125*[4] through the OpenAI API. Our choice was motivated by its advanced capabilities in generating human-like text. We also add a $summary\_percentage$ parameter which specifies the desired reduction in length. For instance, a $summary\_percentage$ of 30 indicates that the summary will be 30% of the original text size.

Importance Ranking. Importance ranking identifies the most salient sentences in a text. The first

[3] https://github.com/miso-belica/sumy

[4] https://platform.openai.com/docs/models/gpt-3-5-turbo

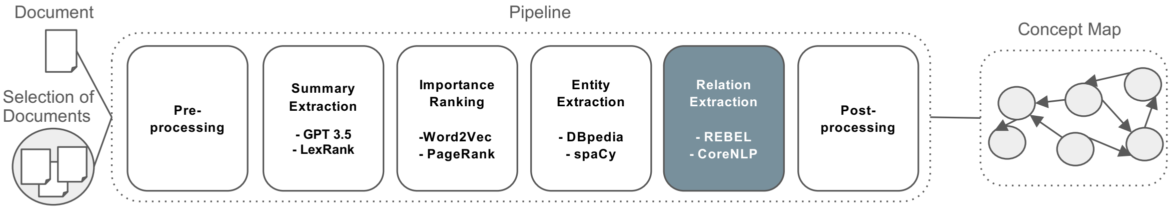


Figure 2: Overview of our pipeline method for automatic concept map creation from a single document or a collection of documents. The pipeline contains one mandatory part highlighted in dark, relation extraction, while the other modules are optional

technique is based on Word2Vec (Mikolov et al., 2013)[5]. We used the standard measure of cosine similarity to assess the relatedness between two sentences. Sentences which are similar to many other will be ranked the highest, as such sentences are likely to convey the most important messages in the text (Cheng and Lapata, 2016). The second one is PageRank (Page et al., 1999) which was selected due to its establishment as a baseline in the prior research of Falke et al. (2017), in line with the intuition that a page's rank should be high when the cumulative ranks of the inbound edges pointing to it are also high. Similarly to the summarization component, we also add as parameter a $ranking\_perc\_threshold$ to select the top sentences scored in the ranking phase.

Entity Extraction. Entity extraction is used to extract relevant entities from text. We either used DBpedia Spotlight (Mendes et al., 2011) with a confidence score of 0.7, or noun chunks from spaCy[6].

Relation Extraction. For relation extraction we used two methods. First, as in Huguet Cabot and Navigli (2021), we refer to relation extraction as the task of extracting triples $(subject, predicate, object)$ from text, with no given entity spans. For this sub-component we fine-tuned REBEL (Huguet Cabot and Navigli, 2021), an open-source, triple extraction, sequence-to-sequence model based on BART (Lewis et al., 2019). The choice of REBEL relies on its state-of-the-art performance across multiple tasks and limited number of parameters compared to other state-of-the-art systems such as UniREl (Tang et al., 2022) or DEEPSTRUCT (Wang et al., 2022). For a comparison with a relation extraction system more similar to the one used by the state-of-the-art, we also included CoreNLP[7] as an alternative.

Post-processing. We implement a post-processing step to identify and remove redundant triples. This step checks for overlapping elements within the triples and removes any triple that overlaps more than 60% with another.

3.2 LLMs-based end-to-end method

LLMs perform better when tasks are decomposed into smaller chunks (Wei et al., 2022). To explore this, we compare three approaches in increasing order of complexity: (I) zero-shot, (II) one-shot, and (III) decomposed prompting. Each approach incrementally adds context and guidance to enhance performance. For (I) and (II), we used similar prompts, with the key difference being that the one-shot prompting (II) includes an example concept map from the training corpus. (III) aims to divide a complex task into simpler sub-tasks for a more efficient prompting, and outperforms standard prompting baselines in complex tasks (Khot et al., 2023). Figure 3 shows the sub-tasks we added in our decomposed prompting baseline. We also provide notebooks to experiment with the LLMs baselines[8].

4 Experimental Setup

4.1 Data and Baselines description

We used the WIKI English dataset (Falke, 2019) for CM-MDS, which was obtained through an automated corpus extension method that combines automatic pre-processing, crowd-sourcing, and expert annotations. It contains 38 clusters, each with several documents and centered on a distinct topic. It is split 50/50 across the train and the test set. Each cluster contains 15 documents in average, and comes with a reference concept map. This dataset

[5] https://radimrehurek.com/gensim/models/word2vec.html

[6] https://spacy.io/usage/linguistic-features

[7] https://github.com/stanfordnlp/CoreNLP

[8] https://github.com/vs1rr/automatic_concept_map_extraction/tree/master/notebooks

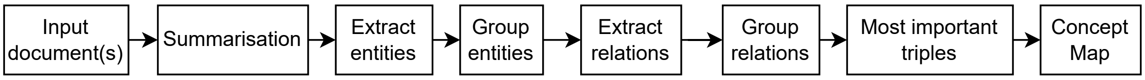


Figure 3: Prompts used for the decomposed prompting baseline.

is the largest annotated corpus for CM-MDS, followed only by the EDUC dataset (Falke, 2019), which contains 30 document clusters focused solely on educational content. Given the unavailability of EDUC dataset, we only evaluated our model on WIKI for the CM-MDS task.

We compare our model against supervised and unsupervised methods proposed in the literature. Unsupervised methods include Page et al. (1999), Leake (2006), Žubrinić et al. (2015). Supervised methods include Falke and Gurevych (2017), and Falke et al. (2017). Lastly, we compare our model to our three end-to-end LLM baselines.

4.2 Fine-tuning REBEL

Falke et al. (2017) used the BIOLOGY (Olney et al., 2011) dataset to evaluate their relation extraction approach, and the WIKI (Falke, 2019) dataset to evaluate their pipeline end-to-end. BIOLOGY contains manually constructed concept maps developed in the work of Olney et al. (2011) and aligned with their corresponding original text by Falke et al. (2017)[9]. Similarly to them, we fine-tune REBEL using the relations from BIOLOGY. Focusing on relations extracted from a single document simplifies the mapping process, as it is easier to associate one sentence to a relation within a single context rather than across multiple documents, therefore we only considered BIOLOGY for the fine-tuning.

We mapped each relation in a concept map to the sentence in the text containing that relation, since relation extraction operates at the individual sentence level. We implemented a rule-based system that returns a boolean value of whether the information in the input triple is present in input the sentence. This process resulted in 220 mappings which we divided into training, evaluation, and test sets for fine-tuning. The split for train/eval/test was 80/10/10. We used the following parameters: $learning\_rate = 2.5 * 10^{-5}$, $epochs = 10$, $batch\_size = 4$, $seed = 1$. We compare the base REBEL to our fine-tuned REBEL.

4.3 Evaluation Metrics

For the evaluation of our results, we use the same metrics as Falke (2019): adapted versions of METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2004)[10] for automatic concept map evaluation. The original metrics are standardly used for machine translation evaluation and automatic summarization, and do not take into consideration any graph related parameters. Additional computational details about these metrics can be found in Appendix A.

4.4 Parameters

We ran our experiments on one Ubuntu machine with 2 GPUs, 40 CPUs, and 348 GiB of memory. The experiments took around 1 day to run.

For the summarization part, we solely focused on document-level summarization, instead of cluster-level summarization. We used *gpt3.5-turbo-0125* and set a $temperature$ of 0, to keep the summary as close as possible to the original text. To avoid repeatedly calling the OpenAI API, we pre-cached the summaries to make our method cost-efficient. For the entity extraction, we set up a local DBpedia Spotlight API[11] and used *en_core_web_lg* for the spaCy model. For the relation extraction, we used an openly available REBEL tokenizer[12].

4.5 Hyperparameter tuning

We first experimented on WIKI train to select the most meaningful parameters for the pipeline. Table 2 shows the different parameters that were tested. For the summary and the ranking part, we investigated the impact of *method* and *percentage* on the quality of the concept maps. For the entity extraction, the two methods were DBpedia Spotlight (*ds*) or the spaCy noun chunks (*nps*). For the relation part, we compared the regular REBEL model (*rebel_hf*) to its fine-tuned version (*rebel_ft*) and *corenlp*. We lastly added some ablation studies where we integrate only the summary part or the

[9] BIOLOGY was accessed with permission from the authors. Due to ownership constraints, the link to the dataset cannot be provided.

[10] We used METEOR 1.5 and ROUGE 1.5.5.

[11] https://github.com/MartinoMensio/spacy-dbpedia-spotlight

[12] https://huggingface.co/Babelscape/rebel-large

ranking part with entity and relation extraction.

The different sets of parameters resulted in 144 different parameter combinations. We found the following parameters yielded the highest scores: $summary\_method = chat\_gpt$, $summary\_percentage = 15$, $ranking\_how = all$, $ranking\_perc\_threshold = 15$, $entity = dbpedia\_spotlight$ and $relation = rebel\_hf$. We use $ranking = page\_rank$ and $ranking = word2vec$ for summarisation and importance ranking, and ranking only respectively. More details on the results of these experiments can be found in Appendix B.

Table 2: Parameter values for each component. *rebel_hf*: base REBEL model, *rebel_ft*: fine-tuned REBEL model, *ds*: DBpedia Spotlight, *nps*: noun chunks from spaCy. Bolded values are the ones kept for the final results.

Component	Parameters	Values
Summary	*method* *percentage*	***chat-gpt***, *lex-rank* **15**, 30
Ranking	*method* *percentage*	***word2vec***, ***page_rank*** **15**, 30
Entity	*method*	***ds***, *nps*
Relation Extraction	*method*	***rebel_hf***, *rebel_ft*, *corenlp*

5 Results

5.1 Quantitative Results

Table 3 shows results for both training and test sets of WIKI, across different component combinations: Full Pipeline, i.e. the one with all the components (A), Pipeline without Ranking (B), Pipeline without Summary (C), and Pipeline without Entities (D) and for the LLM-s. In the next subsections, we will describe the each result for each pipeline in details.

5.1.1 Pipeline approach

(A) demonstrates competitive performance across multiple evaluation metrics on both the training and the test sets. It achieves an F1 score of 26.65 for METEOR on the training set and 24.05 on the test set, outperforming the previous state-of-the-art (Falke et al., 2017). The pipeline achieves recall ROUGE-2 scores consistent with existing literature, attaining an F1 score of 10.64 on the training set and 7.61 on the test set. However, the Precision score for ROUGE-2 remains lower. This suggests that while the pipeline seem to produce comprehensive concept maps that capture a wide range of information, it may also introduce words and details not present in the reference concept maps.

5.1.2 Ablation Studies

In this section, we analyze the significance of each component in the Full Pipeline. Generally, incorporating either the ranking or summary modules led to improved METEOR performance. Comparing METEOR metrics from (B) and (C) to those of (A) reveals an improvement of approximately 10 points for precision, while results for recall and F1 are more mitigated. Omitting the ranking module in (B) resulted in a decline in ROUGE-2 scores (F1 of 3.84 instead of 7.61 in (A)), whereas excluding the summary module in (C) showed a decrease in METEOR scores (F1 of 22.16 instead of 24.05 in (A)). The Pipeline without Entities Extraction (D) shows that the performance drops in METEOR Precision and ROUGE-2 Recall, indicating a lower accuracy when generating the concept map and increased noise. These issues may arise because the model, lacking entity constraints, tends to extract irrelevant triples, leading to less precise and less comprehensive summaries. More details on the hyperparameters are presentend in Appendix B.

5.1.3 LLM end-to-end approach

Across all the three LLMs baselines, METEOR generally shows higher scores compared to ROUGE-2, suggesting that the generated summaries are evaluated more favorably based on linguistic quality metrics rather than exact overlap. The LLMs approach's challenge in achieving high ROUGE-2 Precision suggests that while the generated concept map captures crucial information, it faces difficulty in precisely selecting and summarizing essential details without including redundant or unnecessary information. Decomposed prompting consistently outperforms the two other baselines in METEOR scores and ROUGE-2 Recall on both the training and the test set with respective scores of 28.5 and 24.3.

Across all four pipelines, including (A), ROUGE-2 scores consistently lagged behind the existing literature baselines, particularly in precision, highlighting potential limitations in capturing all pertinent details despite effectively conveying main points, as indicated by higher METEOR scores. This suggests opportunities for enhancing content coverage and lexical alignment. The higher ROUGE-2 recall metrics observed in (C), which excludes summarization, may highlight challenges

Table 3: Results for all systems on WIKI TRAIN and WIKI TEST."-" indicates that we couldn't access to the results. Bolded and underlined metrics are the highest and the second-highest in the column respectively.

Approach	**WIKI TRAIN**						**WIKI TEST**					
	METEOR			ROUGE-2			METEOR			ROUGE-2		
	Pr	**Re**	**F1**	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**
Page et al. (1999)	-	-	-	-	-	-	13.27	14.13	13.62	8.35	6.17	7.01
Leake (2006)	-	-	-	-	-	-	13.44	13.79	13.55	8.57	7.16	7.61
Žubrinić et al. (2015)	-	-	-	-	-	-	14.63	14.92	14.72	10.50	7.91	8.87
Falke and Gurevych (2017)	-	-	-	-	-	-	14.30	23.11	17.46	6.77	23.18	10.20
Falke et al. (2017)	-	-	-	-	-	-	19.57	18.98	19.18	**17.00**	10.69	**12.91**
(A) Full Pipeline	27.08	**28.6**	26.65	**9.67**	13.97	**10.64**	24.61	**24.47**	24.05	6.37	11.81	7.61
Zero-shot Prompting	24.98	20.20	21.36	7.74	16.00	9.05	25.18	19.11	21.24	6.28	15.93	8.22
One-shot Prompting	26.69	21.38	22.64	6.22	19.16	8.40	25.15	19.15	21.26	6.31	15.89	8.24
Decomposed Prompting	**39.9**	25.2	**30.0**	4.8	**27.5**	7.3	**38.4**	23.3	**28.5**	3.9	**24.3**	6.0
Ablation studies												
(B) Pipeline without Ranking	34.59	23.06	26.96	3.17	23.68	5.43	35.91	20.6	25.69	2.16	22.96	3.84
(C) Pipeline without Summary	35.31	20.43	25.37	2.08	23.7	3.76	36.39	16.17	22.16	1.33	24.26	2.5
(D) Pipeline without Entities	27.75	25.58	25.67	7.73	14.6	9.48	24.91	21.59	22.74	4.94	11.92	6.52

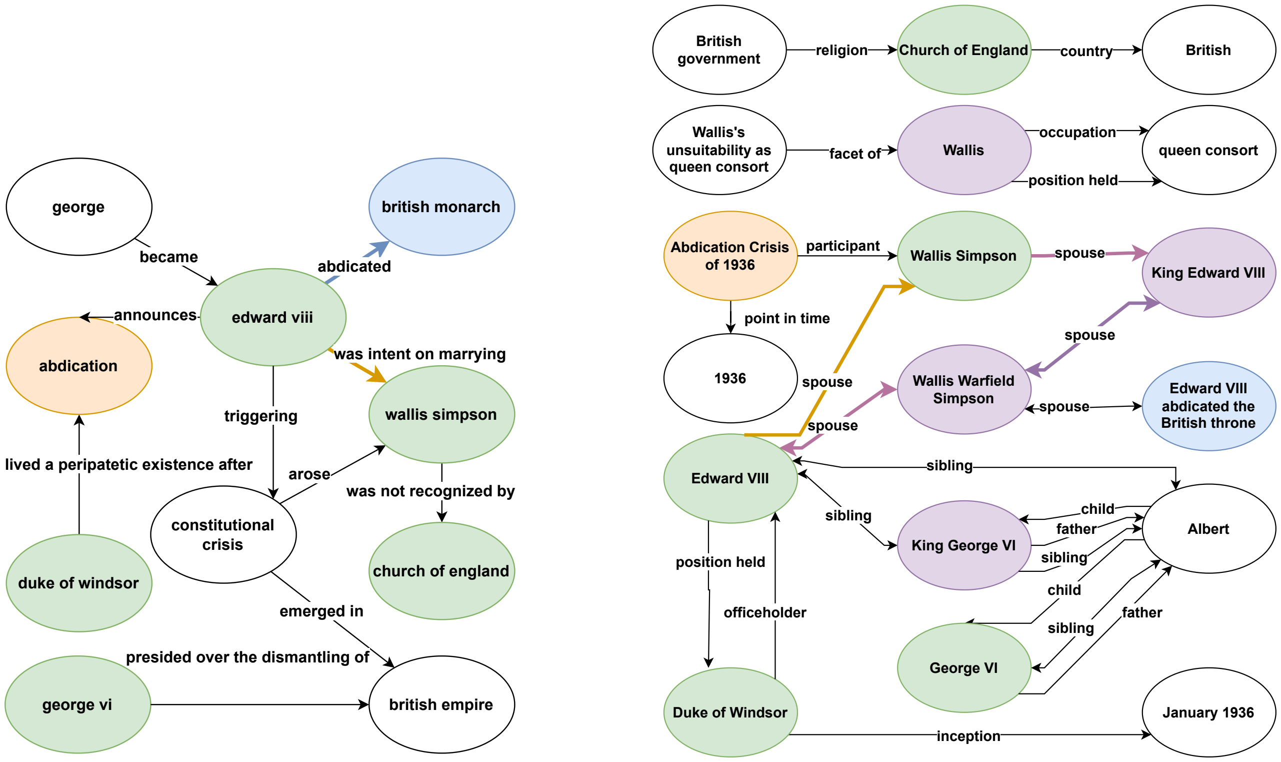
in the summarisation processes which can lead to occasional inaccuracies.

5.2 Qualitative Analysis

We compare the gold-standard concept map of folder 320, as provided by the WIKI dataset, and the output concept map of our full pipeline method. We use color-coding to identify the mappings on nodes and edges, as depicted in Figure 4. On matching nodes to nodes and edged to edges, we introduce the green and orange color. The green color refers to exact match on node or edge level, the orange color represents semantically similar nodes or edges between the gold standard and our concept map. With blue color we represent the node in our concept map that is similar to parts of the gold-standard, more specifically the node "Edward VIII abdicated the British throne" which is similar to (node: edward viii – edge: abdicted – node: british monarch). The purple color groups nodes and edges that are semantically similar in our concept map. In the comparison between the gold-standard and our concept map, we do not find any associations with contradictory meaning, such in case ($node_i$ – edge: parent – $node_j$) and ($node_i$ – edge: child – $node_j$).

On one side, we observe that our pipeline was able to generate a concept map similar to the gold standard where the main concepts are the same, colored in green, or semantically similar, colored in orange and blue. For this example the main concept our pipeline missed is the node: "constitutional crisis". Although "george" and "british empire" are also not present in our approach, we argue that they refer to similar parts in our concept map such as the nodes: "King George VI" and "British". Furthermore, we notice that our generated concept map produces many semantically similar nodes. These nodes are colored in purple in Figure 4b, such as: "King George VI" and "George VI", and "Walls", "Wallis Warfield Simpson", and "Walls Simpson". The co-reference resolution of the concepts will increase the pipeline's performance. On the other side, the relations between nodes appears to be a more challenging task. Our generated concept map was able to produce only a small number of corresponding edges. One explanation might be the complex nature of multiple associations between the main concepts in the documents, as the main concepts have often multiple relations between them. An example can be a wife and a husband nodes that share multiple relations between them such as that they are married, and the multiple common actions they take together.

Lastly, an important aspect to consider is the size of the concept map. Our approach generates on average 32 triples, while the WIKI dataset gold-standard concept maps have on average 12 statements for each multi-document folder. The lower number of statements could be a critical limitation to the expressive freedom of concept maps in terms of content and structure, which is a crucial aspect of the quality of the concept maps (Cañas et al., 2012). Furthermore, the restrictions on size are not communicated beforehand, which might be a factor



(a) Gold-standard.

(b) Pipeline generated.

Figure 4: The comparison of concept maps between the gold-standard and the pipeline generated one based on the folder 320 of WIKI dataset.

reflected in the evaluation metrics of our model that misses excellent performance.

6 Limits & Concerns

Our pipeline architecture demonstrate competitive capabilities compared to baselines, yet also presents areas for improvement. First, the triples extracted appear to be of good quality, but lower ROUGE-2 scores suggest possible omissions in our system's output. However, this may also indicate stronger summarization performance. Second, reproducing results with OpenAI models can be challenging and inconsistent, even when using the same summaries from our experiments. To mitigate potential issues like hallucinations, we consistently set the temperature to 0 when employing OpenAI models. Lastly, evaluating beyond quantitative metrics poses challenges but is essential for a comprehensive assessment. This is why, we performed a first qualitative analysis. In future work, conducting thorough analyses using ROUGE metrics can enhance the quality and accuracy of our results by penalizing hallucinatory outputs.

7 Conclusion & Future Work

We present a novel, open-access and modular pipeline for automated concept map extraction from text. Our system is composed of the following components: summarization of the original input document, importance ranking, entity extraction and relation extraction. We fine-tune a sequence-to-sequence model for relation extraction. We compare our method against our three end-to-end LLMs baselines. The decomposed prompting method yielded the best results for METEOR F1 scores and ROUGE-2 Recall, demonstrating superior performance. Furthermore, the decomposed prompting approach surpasses the current state of the art for METEOR F1 scores and for ROUGE-2 Recall, competing with both supervised and unsupervised methods. In future work, given the current lack of domain-specific evaluation metrics for concept maps, we aim to develop a new metric that integrates graph structure and characteristics along with human feedback tailored to specific needs of concept map creation.

References

David Paul Ausubel, Joseph Donald Novak, Helen Hanesian, et al. 1968. *Educational psychology: A cognitive view*, volume 6. holt, rinehart and Winston New York.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Merve Bayrak and Deniz Dal. 2024. A new methodology for automatic creation of concept maps of turkish texts. *Language Resources and Evaluation*, pages 1–38.

Alberto J Canas, Kenneth M Ford, Joseph D Novak, Patrick Hayes, et al. 2001. Online concept maps. *The Science Teacher*, 68(4):49.

Alberto J Cañas, Joseph D Novak, and Priit Reiska. 2012. Freedom vs. restriction of content and structure during concept mapping-possibilities and limitations for construction and assessment.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494.

S Chitrakala, N Moratanch, B Ramya, CG Revanth Raaj, and B Divya. 2018. Concept-based extractive text summarization using graph modelling and weighted iterative ranking. In *emerging research in computing, information, communication and applications: ERCICA 2016*, pages 149–160. Springer.

Camila de Aguiar, Davidson Cury, and Amal Zouaq. 2016. Automatic construction of concept maps from texts. pages 1–6.

Douglas D. Dexter and Charles A. Hughes. 2011. Graphic organizers and students with learning disabilities: A meta-analysis. *Learning Disability Quarterly*, 34(1):51–72.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Tobias Falke. 2019. *Automatic Structured Text Summarization with Concept Maps*. Ph.D. thesis, Technische Universität, Darmstadt.

Tobias Falke and Iryna Gurevych. 2017. Bringing structure into summaries: Crowdsourcing a benchmark corpus of concept maps. *arXiv preprint arXiv:1704.04452*.

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 Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 801–811.

Tobias Falke, Gabriel Stanovsky, Iryna Gurevych, and Ido Dagan. 2016. Porting an open information extraction system from english to german. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 892–898.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks.

Juliana H Kowata, Davidson Cury, and Maria Claudia Silva Boeres. 2010. Concept maps core elements candidates recognition from text. In *Proceedings of Fourth International Conference on Concept Mapping*, pages 120–127.

Alejandro Valerio David Leake. 2006. Jump-starting concept map construction with knowledge extracted from documents. In *In Proceedings of the Second International Conference on Concept Mapping (CMC*, pages 296–303.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

John C Nesbit and Olusola O Adesope. 2006. Learning with concept and knowledge maps: A meta-analysis. *Review of educational research*, 76(3):413–448.

Joseph D Novak. 1990. Concept maps and vee diagrams: Two metacognitive tools to facilitate meaningful learning. *Instructional science*, 19(1):29–52.

Joseph D Novak and D Bob Gowin. 1984. *Learning how to learn*. cambridge University press.

AB Nugumanova, Aizhan Soltangalienva Tlebaldinova, Ye M Baiburin, and Ye V Ponkina. 2021. Natural language processing methods for concept map mining: The case for english, kazakh and russian texts. *Journal of Mathematics, Mechanics and Computer Science*, 112(4).

Ana Oliveira, Francisco Câmara Pereira, and Amílcar Cardoso. 2001. Automatic reading and learning from text. In *Proceedings of the international symposium on artificial intelligence (ISAI)*. Citeseer.

Andrew Olney, Whitney L Cade, and Claire Williams. 2011. Generating concept map exercises from textbooks. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 111–119.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Iqbal Qasim, Jin-Woo Jeong, Jee-Uk Heu, and Dong-Ho Lee. 2013. Concept map construction from text documents using affinity propagation. *Journal of Information Science*, 39(6):719–736.

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.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. UniRel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jorge Villalon. 2012. *Automated Generation of Concept Maps to Support Writing*. University of Sydney.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. Deepstruct: Pretraining of language models for structure prediction. *arXiv preprint arXiv:2205.10475*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Amal Zouaq, Dragan Gasevic, and Marek Hatala. 2011. Ontologizing concept maps using graph theory. In *Proceedings of the 2011 ACM Symposium on applied computing*, pages 1687–1692.

Krunoslav Zubrinic, Damir Kalpic, and Mario Milicevic. 2012. The automatic creation of concept maps from documents written using morphologically rich languages. *Expert systems with applications*, 39(16):12709–12718.

Krunoslav Žubrinić, Ines Obradović, and Tomo Sjekavica. 2015. Implementation of method for generating concept map from unstructured text in the croatian language. In *2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 220–223. IEEE.

A Evaluation Metrics

For the METEOR-adapted metric, we compute Precision, Recall as described in Falke et al. (2017). Given two pair of propositions $\mathbf{p_s} \in \mathbf{P_S}$ and $\mathbf{p_r} \in \mathbf{P_R}$, where P_R and P_S are the set of triples from the reference and from the system respectively, we calculate the match score $\mathbf{meteor}(\mathbf{p_s}, \mathbf{p_r}) \in [\mathbf{0}, \mathbf{1}]$. Precision and Recall are then computed as in Falke et al. (2017) as:

$$Pr = \frac{1}{|P_S|} \sum_{\mathbf{p} \in \mathbf{P_S}} \max\{\text{meteor}(\mathbf{p}, \mathbf{p_r}) | \mathbf{p_r} \in \mathbf{P_R}\}$$
$$Re = \frac{1}{|\mathbf{P_R}|} \sum_{\mathbf{p} \in \mathbf{P_R}} \max\{\text{meteor}(\mathbf{p}, \mathbf{p_s}) | \mathbf{p_s} \in \mathbf{P_S}\}$$

The ROUGE-2-based Precision and Recall were computed as in Falke et al. (2017), by merging all propositions within a map into two separate strings, $\mathbf{s_s}$ and $\mathbf{s_r}$. Following Falke et al. (2017), the F1-score represents the balanced harmonic average of Precision and Recall. Scores for each concept map are macro-averaged across all topics.

B Hyperparameter tuning

We describe the main results of the hyperparameter tuning, and make the full results with metrics available together with our code[13].

Table 4 shows the correlation between the entity and relation features and the averaged F1 score between the METEOR F1 and the ROUGE F1. It can be seen that DBpedia Spotlight (*ds*) performs significantly better than noun chunks from spaCy (*nps*) for entity extraction. For relation extraction, *rebel_hf* and *rebel_ft* perform significantly better than *corenlp*, but there is no strong differences between the two REBEL models. The correlations for averaged precision and recall were -0.10 ($pval = 0.33$) and -0.20 ($pval = 0.05$) respectively. We therefore chose $entity = dbpedia\_spotlight$ and $relation = rebel\_hf$ for the entity and relation extraction parameters.

Table 4: Correlation between entity + relation features and average F1 scores between METEOR and ROUGE. The table reads as follows: a correlation of -0.64 for avg_f1 means that there is a negative correlation between *ds* entity and avg_f1, compared to *nps* entity.

Feature	Value 1	Value 2	Metric	Correlation	P-value
entity	***ds***	*nps*	avg_f1	-0.64	$2.10e-17$
$relation$	*corenlp*	***rebel_ft***	avg_f1	0.37	$2.63e-4$
$relation$	*corenlp*	***rebel_hf***	avg_f1	0.38	1.34e-4
$relation$	*rebel_hf*	*rebel_ft*	avg_f1	0.012	0.90

Table 5: Correlation between features and F1 scores. S: System. For the features (F): S: summary method, SP: summary percentage, IR: importance ranking, IRP: importance ranking percentage. Bolded correlations are the ones that are statistically significant ($pval < 0.05$) and higher in absolute value than 0.1.

S	F	Value 1	Value 2	Metric	Correlation	P-value
A	S	*chat-gpt*	*lex-rank*	avg_f1	**-0.92**	$5.51e-7$
				avg_pr	**-0.56**	0.03
				avg_re	**-0.63**	$8.98e-3$
	SP	15	30	avg_f1	-0.05	0.85
				avg_pr	0.21	0.44
				avg_re	0.40	0.12
	IR	*page_rank*	*word2vec*	avg_f1	-0.14	0.82
				avg_pr	-0.08	0.76
				avg_re	-0.15	0.57
	IRP	15	30	avg_f1	-0.06	0.82
				avg_pr	0.057	0.02
				avg_re	0.45	0.079
B	S	*chat-gpt*	*lex-rank*	avg_f1	**-0.96**	0.037
				avg_pr	0.55	0.45
				avg_re	-0.50	0.50
	SP	15	30	avg_f1	-0.26	0.74
				avg_pr	0.67	0.33
				avg_re	0.71	0.29
C	IR	*page_rank*	*word2vec*	avg_f1	-0.89	0.11
				avg_pr	-0.36	0.64
				avg_re	-0.60	0.40
	IRP	15	30	avg_f1	-0.37	0.63
				avg_pr	0.93	0.069
				avg_re	0.80	-0.37

We then only kept the experiments that used DBpedia Spotlight for entity extraction and $rebel\_hf$ for relation extraction. We looked at the best parameters for summarisation and importance ranking for each type of system independently: (A) Full Pipeline (B) Full Pipeline without Ranking, and (C) Full Pipeline without Summary.

Table 5 shows the correlations between each feature in the three systems and the average F1, Precision and Recall scores. The only correlation that is higher than 0.1 and statistically significant is the one comparing the summarisation methods: *chat-gpt* performs significantly better than *lex-rank*. Since the other results had weak or non-significant correlations, we chose the parameters that got the highest averaged F1 scores on the WIKI train dataset:

- A: $summary\_method = chat\_gpt$, $summary\_percentage = 15$, $ranking = word2vec$ and $ranking\_perc\_threshold = 15$.
- B: $summary\_method = chat\_gpt$, $summary\_percentage = 15$.
- C: $ranking = page\_rank$ and $ranking\_perc\_threshold = 15$.

[13] The CSV with the completed results can be found here.

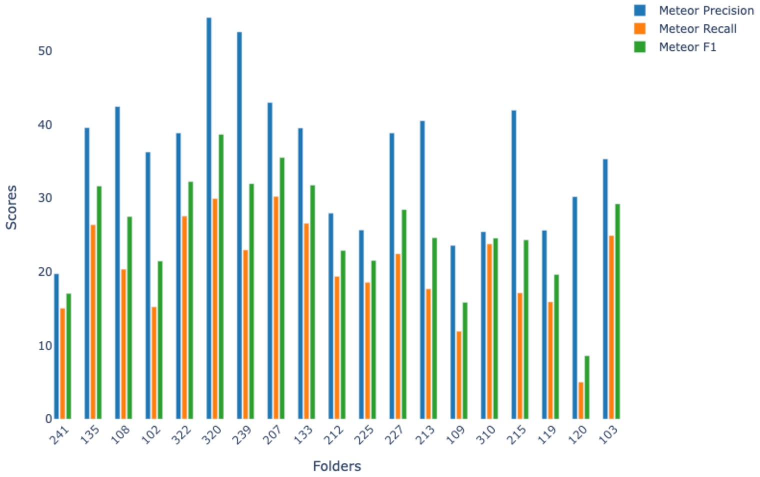


Figure 5: The plot displays METEOR Precision, Recall, and F1 results on the test set for each multi-document concept map, using the parameters: $chat - gpt$ summarization with a 15% summary percentage, and a non-fine-tuned Rebel model.

C Qualitative Example Characteristics

We chose our example on the basis of their METEOR and ROUGE-2 average scores. Our choice is on folder 320 of the test set which had the following METEOR values: Precision= 42.60, Recall = 41.84 and F1 = 42.26, and the following ROUGE-2 values: Precision = 8.60, Recall = 25.39 and F1 = 12.85.