Automated Concept Map Extraction from Text

Martina Galletti[1,2,*], Inès Blin[1,3,*], Eleni Ilkou[4]

[1] Sony Computer Science Laboratories - Paris, 6 Rue Amyot, 75005, Paris, France,
[2] Sapienza University of Rome, Italy
[3] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands,
[4] L3S Research Center, Leibniz University Hannover, Germany
Correspondence: {martina.galletti, ines.blin}@sony.com

Abstract

Concept Maps are semantic graph summary representations of relations between concepts in text. They are particularly beneficial for students with difficulty in reading comprehension, such as those with special educational needs and disabilities (Galletti et al., 2022; Dexter and Hughes, 2011). Currently, the field of concept map extraction from text is outdated, relying on old baselines, limited datasets, and limited performances with F1 scores below 20%. We propose a novel neuro-symbolic pipeline and a GPT3.5-based method for automated concept map extraction from text evaluated over the WIKI dataset. The pipeline is a robust, modularized, and open-source architecture, the first to use semantic and neural techniques for automatic concept map extraction while also using a preliminary summarization component to reduce processing time and optimize computational resources. Furthermore, we investigate the large language model in zero-shot, one-shot, and decomposed prompting for concept map generation. Our approaches achieve state-of-the-art results in METEOR metrics, with F1 scores of 25.7 and 28.5, respectively, and in ROUGE-2 recall, with respective scores of 24.3 and 24.3. This contribution advances the task of automated concept map extraction from text, opening doors to wider applications such as education and speech-language therapy. The code is openly available[1].

1 Introduction

Concept Maps 3.0 (Jensen and Johnsen, 2016) leverage semantic web (SW) technologies to create dynamic concept maps (CMs). These summaries of visual graphs represent the semantic relationships between concepts extracted from text, as shown in the concept map extracted in Table 1 and visualised in Figure 1. CMs are widely used in education and speech and language therapy (Villalon, 2012).

Table 1: Example of a concepts map extraction from folder 320 of WIKI (Falke, 2019).

Reference Concept Map
(constitutional crisis, emerged in, british empire)
(constitutional crisis, arose ,wallis simpson)
(duke of windsor, lived a peripatetic existence after, abdication)
(edward viii, announces, abdication)
(edward viii, abdicated, british monarch)
(edward viii, triggering, constitutional crisis)
(edward viii, was intent on marrying, wallis simpson)
(george, became, edward viii)
(george vi, presided over the dismantling of, british empire)
(wallis simpson, was not recognized by, church of england)

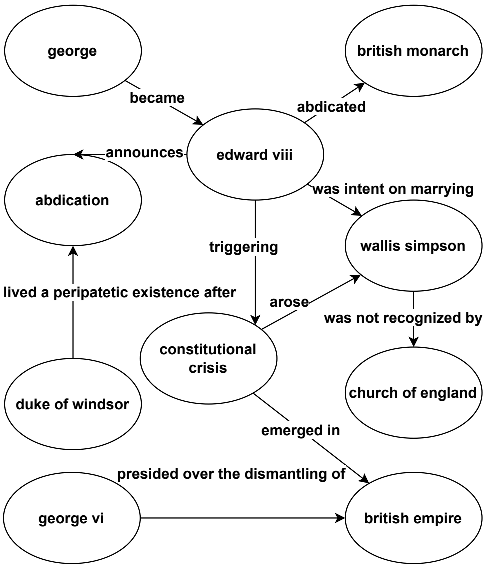


Figure 1: The visualisation of Concept Map of Table 1.

They facilitate the integration of new information with old knowledge (Canas et al., 2001), promote active processing of information (Novak, 1990), improve long-term memory retention, and foster better understanding and critical thinking (Novak and Gowin, 1984).

SW technologies have proven highly effective when integrated with CMs in various applications. For example, ontology-based approaches provide a structured approach to knowledge representation,

[*] These authors contributed equally.

[1] https://github.com/SonyCSLParis/concept_map

allowing the generation of CMs (Verhodubs and Grundspenkis, 2013). These technologies have also been used to automate the CM scoring (Park and Calvo, 2008), optimizing the evaluation process. In addition, tools such as Semantic MediaWiki (Krötzsch et al., 2006) have been incorporated into CMs to support collaborative ontology maintenance (Hedayati et al., 2017). In educational contexts, the synergy between CMs and SW technologies has been instrumental in the development of ontologies that support adaptive learning systems (Chu et al., 2011; Icoz et al., 2014). This combination provides a powerful tool for representing and organizing knowledge, enabling the creation of shareable educational resources and improving the interoperability and accessibility of educational resources (Jiang et al., 2008).

CMs are powerful tools that can improve comprehension and learning, as they provide users with a structured way to organize and visually represent knowledge, making complex content more accessible (Ausubel et al., 1968; Nesbit and Adesope, 2006; Dexter and Hughes, 2011). More specifically, grasping the meaning of entire texts can be frustrating and exhausting for students with special educational needs and disabilities, such as those diagnosed with reading comprehension disorders. Furthermore, CM applications extend beyond learning and rehabilitation, as shown by studies in information retrieval and knowledge representation (Villalon, 2012; Cañas and Novak, 2006).

The manual creation of CMs from text is challenging and impractical due to the time-consuming nature of the task. As a result, attention has been paid to the automatic extraction of CMs from text (de Aguiar et al., 2016; Falke, 2019). However, existing methods are outdated, with Falke et al. (2017) being the latest state-of-the-art (SOTA) method with F1 performance of 19.18 and 12.91 for METEOR and ROUGE-2, respectively. These methods rely solely on symbolic or machine learning approaches, excluding neural methods. They typically consist of pipelines that integrate components such as entity and relation extraction. Moreover, they have shortcomings such as limited efficiency in processing large datasets, reliance on annotated datasets for supervised models, and lack of open access to the underlying code.

In this paper, we contribute the following:

- We propose a novel open-access[2] neuro-symbolic pipeline for automatic CM extraction from single and multiple documents. Our approach incorporates a new summarization component that enhances efficiency by a 3-4x speed-up. Moreover, it includes a fine-tuned REBEL model (Huguet Cabot and Navigli, 2021) for this task. When tested for multiple documents, it outperforms previous pipelines on METEOR F1 (24.0%) score;

- We investigate the robustness of the proposed pipeline by removing different semantic modules, and observe the competitive performance of F1 scores for METEOR above 20% across all different methods;

- We investigate the ability of GPT3.5 to be used in end-to-end methods for automated CM extraction. The best performance is achieved with decomposed prompting, with SOTA performance in METEOR Precision (38.4%) and F1 (28.5%), and ROUGE-2 Recall (24.3%).

2 Related Work

Concept Map 3.0 suggests the use of CMs enriched by Web 3.0 technologies, using SW resources, such as schema.org and Wikidata, and following Web Data Principles to make them machine-interpretable and semantic learning resources (Jensen and Johnsen, 2016). Towards this line, we contribute with our neuro-symbolic pipeline for automatic CM extraction that utilizes SW tools. Although this task can be broken down into several components, evaluating these individual components is beyond the scope of this task and of our work. We focus solely on complete approaches for CM extraction from text.

Currently, the literature conventionally portrays automatic CM extraction from text as a multistep approach involving subtasks such as concept and relation extraction and subgraph selection. Existing works are twofold: those with a single document as input, namely the CM - Document Summarization (CM-DS) task (Falke et al., 2017), and those with multiple documents as input, namely the CM - Multi Document Summarization (CM-MDS) task.

For **CM-DS**, Oliveira et al. (2001) laid the foundation not only by extracting relations between concepts from a text file, but also by extrapolating rules about the knowledge at hand. Subsequent studies such as Cañas and Novak (2006) employed unsu-

[2] https://github.com/SonyCSLParis/concept_map

Table 2: Comparison of existing pipeline methods for CM-DS (S) and CM-MDS (M) tasks from text data to our pipeline. For the header: $Lang.$: Language, $Meth.$: Methods, 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, statistical tools (S), neural tools (N). For Summary Extraction (SE): pre: SE occurs before entity and relation extraction, while $post$: SE occurs after.

Authors	**Task**	**Lang.**	**Meth.**	**SE**	**IR**	**EE**	**RE**
Oliveira et al. (2001)	S	EN	L				✓
Rajaraman and Tan (2002)	M	EN	L		✓	✓	
Cañas and Novak (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	LSN	pre	✓	✓	✓

pervised methods with deep syntactic parsing for concept selection. These methods primarily used term frequencies to assign a document to the most probable CM among a set of options, enhancing the precision of concept selection. Kowata et al. (2010) further focused on extracting CMs from Portuguese news articles. This work pioneered the use of a comprehensive pipeline approach that included text segmentation, tokenization, part-of-speech tagging, core element candidate recognition, dependency interpretation, and CM construction. Subsequently, de Aguiar et al. (2016) introduced a sophisticated pipeline approach that integrated grammar rules, co-reference resolution, and concept ranking based on frequency of occurrence. Lastly, Bayrak and Dal (2024) introduced a new heuristic approach to extract CMs from Turkish texts.

For **CM-MDS**, Rajaraman and Tan (2002) pioneered the field by using regular expressions and term frequency-based grouping to build a CM-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 (Fellbaum, 2010), and clustered them. Their approach was integrated into a system and validated through experimental studies. Zouaq et al. (2011) later defined specific patterns on dependency syntax representations to enhance entity extraction. Their work highlighted the usefulness of CM mining in ontology learning. Žubrinić et al. (2015) extended the CM-MDS task by introducing a heuristic approach to summarize CMs from legal documents written in Croatian. This was a significant advance that demonstrated the adaptability of CM-MDS techniques to other languages and domain-specific document types.

Lastly, Falke et al. (2019; 2017; 2017) made significant contributions to the field. Their model leverages predicate-argument structures and automatic models for German and English, achieving SOTA performance until now. Their pipeline includes five steps: (1) concept and relation extraction, from Open Information Extraction (Etzioni et al., 2008); (2) concept mention grouping and labeling with greedy search optimization (3) relation mention grouping, labeling, and selection using lemmatization; (4) importance estimation with a ranking support vector machine; (5) CM construction using integer linear programming (Gomory, 1958). Their English datasets, WIKI (Falke, 2019) and EDUC (Falke and Gurevych, 2017), are the two largest annotated corpora for CM-MDS and serve as the main benchmark for this task. WIKI was obtained through an automated corpus extension method with automatic pre-processing, crowdsourcing, and expert annotations. It contains 38 groups, each with several documents and focused on a different topic. It is split 50/50 across the training and the test set. Each cluster contains on average 15 documents and comes with a CM reference. EDUC contains 30 document clusters on educational content and was created through crowdsourcing; unlike WIKI, the authors had not released this data set for use in this investigation.

Table 2 summarizes the existing methods for CM-DS and CM-MDS. It showcases the evolution from basic term frequency methods to more complex pipelines. Existing approaches rely on symbolic or machine learning methods, lacking the incorporation of advanced neural techniques that can enhance relation extraction accuracy. We fine-tuned the sequence-to-sequence models for the relation extraction part. Additionally, no previous studies have introduced the preliminary summarization components that we use to reduce processing time and optimize computational resources. Our LLM-based methods and modularized pipeline achieve competitive results when compared with the SOTA.

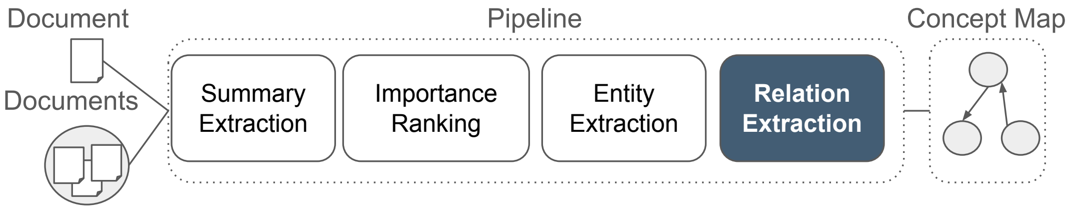


Figure 2: Our pipeline method for automatic CM creation from a single document or a collection of documents. The pipeline contains one mandatory part (in the dark, relation extraction); the other modules are optional.

3 Methods

3.1 Pipeline

We introduce a neuro-symbolic pipeline that is modular and open-access, which consists of four components: (1) the summarization, (2) the importance ranking, (3) the entity extraction, and (4) the relation extraction. The latter component (4) is always required, while the other three can be deactivated, as we show in Figure 2. We are the first to propose (1) as a primary step to reduce processing time and optimize computational resources.

Although we use several well-established components in our pipeline that are not necessarily SOTA in their tasks, our key contribution is the innovative integration of these tools within a cohesive framework for CM extraction. We also investigate whether adding preliminary summarization steps can yield better results by reducing processing time and optimizing computational resources. The preliminary summarization step differs from the SOTA method (Falke et al., 2017), which used graph summarization as the last step.

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 chose this method because it was previously used for concept-based extractive summarization (Chitrakala et al., 2018), and it leverages graph-based and ranking methods that are particularly relevant to 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 to generate human-like text, its strong contextual understanding, and its efficiency in producing coherent and fluent summaries. Compared to earlier models, GPT-3.5 offers improved language generation quality while being more cost-effective than GPT-4, making it well-suited for scalable summarization tasks. Furthermore, its ability to generalize across diverse text domains ensures robustness when applied to complex summarization scenarios. Although we currently use GPT for the three LLM-based models, our approach is not limited to this specific LLM. We also add a *summary_percentage* parameter which specifies the desired reduction in length. For example, 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 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 that are similar to many others 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 is PageRank (Page et al., 1999) which was selected due to its establishment as a baseline in the prior research in 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. The similarity matrix is a square matrix of size (N × N), where N represents the total number of sentences in all summaries within a folder. Each folder contains a concept map derived from multiple documents on the same topic. 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 used DBpedia Spotlight (Mendes et al., 2011) with a confi-

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

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

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

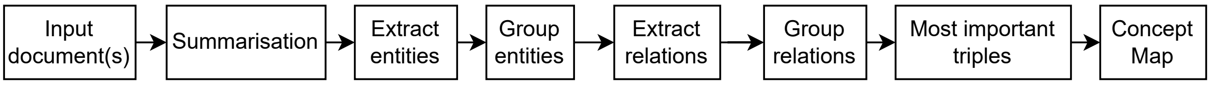


Figure 3: Prompts used for the decomposed prompting approach.

dence score of 0.7, or noun chunks from spaCy[6].

Relation Extraction. 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 subcomponent, 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 is based on its SOTA performance in multiple tasks and a limited number of parameters compared to other SOTA 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 SOTA, we also included CoreNLP[7] as an alternative. Finally, we post-processed the results by removing any triples that overlapped by more than 60% with others, with the aim of eliminating redundancy.

3.2 LLM-based Methods

We investigate the ability of one LLM, the gpt3.5-turbo-0125 (Brown et al., 2020)[8], to generate CMs from text. The LLM tends to perform better when tasks are decomposed into smaller fragments (Wei et al., 2022). We compare three approaches with increasing complexity: (I) "zero-shot", (II) "one-shot", and (III) "decomposed prompt". Each approach incrementally adds context and guidance to enhance performance. For (I) "zero-shot" and (II) "one-shot prompting", we used similar prompts, with the key difference being that the one-shot prompting (II) includes an example CM from the training corpus. The (III) "decomposed prompting" aims to divide a complex task into simpler subtasks for more efficient prompting and outperforms standard prompting baselines in complex tasks (Khot et al., 2023). Figure 3 illustrates the additional subtasks incorporated into our decomposed prompting approach. We focus solely on "zero-shot" settings for each decomposed prompt. Implementing n-shot for each component would have required finer-grained ground truth, such as text summaries or grouped entities, necessitating manual annotation from our side.

We provide notebooks to experiment with the LLM baselines[9], as well as the exact prompt and the code used for all baselines[10] to ensure reproducibility. The only prompt that is reused in our pipeline is the one for summarization. An example of a prompt for the "zero-shot" baseline is shown in Figure 4.

Prompt Zero-Shot Baseline

Task Description: Concept Map Generation Your task is to process a collection of texts and extract triples from them. Subsequently, you'll aggregate this information to construct a unique and comprehensive Concept Map representing the information in all the texts in the given folder. The resulting Concept Map should adhere to the following structure:

```
    <Subject> - <Predicate> - <Object>,
    <Subject> - <Predicate> - <Object>,
    <Subject> - <Predicate> - <Object>,
```

The Concept Map should contain only the most important triple that best summarizes the content of all texts and avoid redundancy across triples. In your answer, you must give the output in a .csv file with the columns "subject", "predicate", and "object". The output is a single .csv file.

Figure 4: The "zero-shot" prompt used for concept map generation.

4 Experimental Setup

4.1 Dataset and Baselines

WIKI (Falke, 2019) and EDUC (Falke and Gurevych, 2017) are the main benchmark datasets in the CM-MDS task. We reached out to the authors for these datasets, and they only provided WIKI, which we use for our experiments on CM-MDS. Expanding our evaluation to other datasets would require access to EDUC or the creation of new datasets, which is beyond the scope of this work. On average for WIKI, the training set has 96

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

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

[8] All experiments were run in May 2024 using GPT-3.5, which was one of the most advanced models available at the time. Since then, newer models have been released. Instead of re-running all experiments, we emphasize the value of our methods rather than model improvements.

[9] https://github.com/SonyCSLParis/concept_map/tree/main/notebooks

[10] https://github.com/SonyCSLParis/concept_map/tree/main/src/baselines

sentences per folder, while the test set has 121 sentences. Although we do not own WIKI, it is easily accessible. With the permission of the owner, we have uploaded it to our GitHub page, ensuring the reproducibility of our work.

We compare our model with supervised (Falke et al., 2017; Falke, 2019) and unsupervised (Page et al., 1999; Cañas and Novak, 2006; Žubrinić et al., 2015) methods from the literature. These baselines are, to the best of our knowledge, the only ones that have reported results on the same corpus and evaluation metrics. Lastly, we compare our model to our three LLM approaches.

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 CMs developed in the work of Olney et al. (2011) and aligned with their original text corresponding to Falke et al. (2017)[11]. 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 fine-tuning.

We map each relation in a CM 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 the input of the sentence. This process was applied to the 183 BIOLOGY documents, resulting in 220 mappings that we divided into training, evaluation, and test sets for fine-tuning. The split for *train / evaluation / 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 in previous work on this task (Falke, 2019): adapted versions of METEOR 1.5 (Banerjee and Lavie, 2005) and ROUGE 1.5.5 (Lin, 2004) for automatic CM evaluation. The original metrics are standardly used for machine translation evaluation and automatic summarization and do not take into consideration graph-related parameters. We selected METEOR and ROUGE-2 over the exact match of F1 because they better capture nuanced overlaps between concepts and relations in CMs. These metrics offer more flexibility, including partial matches and paraphrasing.

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

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

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

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, s_s and s_r. The F1 score represents the balanced harmonic average of Precision and Recall. The scores for each CM are macro averaged across all topics.

4.4 Parameters

We ran our experiments for around 1 day on an Ubuntu machine with 2 GPUs, 40 CPUs, and 348 GiB of memory. For the summarization part, we focused solely on document-level summarization. We used *gpt3.5-turbo-0125* and set a temperature of 0, to keep the summary as close to the original text as possible. To avoid repeatedly calling the OpenAI API, we precached the summaries to make our method cost-efficient. For entity extraction, we set up a local DBpedia Spotlight API[12] and used *en_core_web_lg* for the spaCy model. For relation extraction, we used an openly available REBEL tokenizer[13].

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

Table 3: Parameter values for each component. *rebel_hf* and *rebel_ft*: base and fine-tuned REBEL model, *ds*: DBpedia Spotlight, *nps*: noun chunks from spaCy. Bolded values are used for the final results.

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

4.5 Hyperparameter tuning

We used WIKI TRAIN (Falke, 2019) to select the best parameters for the pipeline, as shown in Table 3. For summary and ranking, we investigated the impact of *method* and *percentage* on the quality of CMs. For 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 make the results available with our code[14].

We analyze the correlation between entity and relation extraction characteristics and the averaged F1 score (computed from METEOR and ROUGE F1). The results show that DBpedia Spotlight (*ds*, encoded as 0) outperforms spaCy's noun chunks (*nps*, encoded as 1) for entity extraction, with a strong negative correlation ($r = -0.64$, $p < 0.05$), indicating that *ds* consistently leads to higher F1 scores. For relation extraction, *rebel_hf* and *rebel_ft* outperform *corenlp*, with moderate positive correlations ($r \approx 0.37$, $p < 0.05$). However, the difference between *rebel_hf* and *rebel_ft* is negligible. We selected *ds* for entity extraction and *rebel_hf* for relation extraction.

We then looked at the best parameters for summarization and importance ranking for each type of system independently: ($\mathscr{A}$) Full pipeline ($\mathscr{B}$) $\mathscr{A}$ without ranking, ($\mathscr{C}$) $\mathscr{A}$ without summary. The only correlation that is statistically significant is the one comparing the summarization methods: *chat-gpt* outperforms *lex-rank*. Since the other results had weak or nonsignificant correlations, we chose the parameters that got the highest averaged F1 scores on the WIKI train dataset. Table 4 shows the final parameters retained.

Table 4: Final parameters retained for each system.

Parameter	$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$
summary_method	*chat-gpt*	*chat-gpt*	-
summary_percentage	15	15	-
ranking	*word2vec*	-	*page_rank*
ranking_perc_threshold	15	-	15

Table 5 shows more detailed results on the correlations between each feature in the three systems and the average F1, Precision, and Recall scores.

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). 'Value 1' is encoded as 0 and 'Value 2' as 1. The correlation of -0.92 in the first row indicates that *avg_f1* tends to be lower when the summarisation method is 0 (*chat-gpt*) rather than 1 (*lex-rank*)..

S	**F**	**Value 1**	**Value 2**	**Metric**	**Correlation**	**P-value**
$\mathscr{A}$	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
$\mathscr{B}$	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
$\mathscr{C}$	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

5 Results

Table 6 shows the results for the training and test sets of WIKI between the baselines of the literature and our methods. We present the results of the pipeline (5.1) and the LLM (5.2) results.

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

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

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

Table 6: Results for all systems on WIKI TRAIN and WIKI TEST. We compare our model against supervised and unsupervised methods from the literature. "-" indicates that we couldn't access to the results. Bolded and underlined metrics are the highest and the second-highest in the column, respectively. $\mathscr{A}$: Full Pipeline ; $\mathscr{B}$: $\mathscr{A}$ without Ranking ; $\mathscr{C}$: $\mathscr{A}$ without Summary. "Zero-shot", "One-shot" and "Decomposed" are prompting techniques.

Approach	**WIKI TRAIN**						**WIKI TEST**					
	METEOR			ROUGE-2			METEOR			ROUGE-2		
	Pr	**Re**	**F1**	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**
Literature baselines												
Page et al. (1999)	-	-	-	-	-	-	13.3	14.1	13.7	8.4	6.2	7.0
Cañas and Novak (2006)	-	-	-	-	-	-	13.4	13.8	13.6	8.6	7.2	7.6
Žubrinić et al. (2015)	-	-	-	-	-	-	14.7	14.9	14.7	10.5	7.9	8.9
Falke and Gurevych (2017)	-	-	-	-	-	-	14.3	23.1	17.5	6.8	23.2	10.2
Falke et al. (2017)	-	-	-	-	-	-	19.6	19.0	19.2	**17.0**	10.7	**12.9**
Pipeline Methods												
$\mathscr{A}$: Full	27.08	**28.6**	26.6	**9.7**	13.9	**10.6**	24.6	**24.5**	24.0	6.4	11.8	7.6
Ablation studies												
$\mathscr{B}$: No Rankings	34.6	23.0	26.9	3.2	23.7	5.4	35.9	20.6	25.6	2.2	22.9	3.84
$\mathscr{C}$: No Summaries	35.3	20.4	25.3	2.0	23.7	3.8	36.4	16.8	22.2	1.3	24.3	2.5
LLM Methods												
Zero-shot	25.0	20.2	21.4	7.7	16.0	9.1	25.2	19.1	21.2	6.3	15.9	8.2
One-shot	26.7	21.4	22.6	6.2	19.2	8.4	25.2	19.2	21.3	6.3	15.9	8.2
Decomposed	**39.9**	25.2	**30.0**	4.8	**27.5**	7.3	**38.4**	23.3	**28.5**	3.9	**24.3**	6.0

5.1 Pipeline

Quantitative Results

Figure 2 illustrates that, in addition to the essential relation extraction step for CM extraction, two other optional core components are summarization and importance ranking. We therefore compare the full pipeline from Figure 2 to combinations removing one of these three components: the one with all the components ($\mathscr{A}$), pipeline without ranking ($\mathscr{B}$), pipeline without summary ($\mathscr{C}$). $\mathscr{A}$ demonstrates competitive performance across multiple evaluation metrics on both the training and test sets. It achieves an F1 score of 26.65 for METEOR in the training set and 24.05 on the test set, outperforming the previous SOTA (Falke, 2019). $\mathscr{A}$ achieves a ROUGE-2 recall score (11.81) consistent with existing literature, but lower F1 scores for the training (10.64) and test (7.61) sets.

Our pipeline produces comprehensive CMs that capture a wide range of information (Lavie and Denkowski, 2009), as the decent scores in METEOR suggest across the four pipelines. Comparing the METEOR metrics from $\mathscr{B}$ and $\mathscr{C}$ to those of $\mathscr{A}$ reveals an improvement of approximately 10 points in precision, while the results for recall and F1 are more mitigated. Excluding the summary module in $\mathscr{C}$ showed a decrease in METEOR scores (F1 of 22.16 instead of 24.05 in $\mathscr{A}$). $\mathscr{B}$ achieves the best F1 METEOR performance, slightly higher than $\mathscr{A}$ where combining summarization and ranking may become too reductive.

The lower ROUGE-2 scores suggest that the pipeline's generated CMs do not include the exact words to match the bigrams of the gold standard, leading to a loss in performance (ShafieiBavani et al., 2017). Omitting the ranking module in $\mathscr{B}$ resulted in a decrease in ROUGE-2 scores (F1 of 3.84 instead of 7.61 in $\mathscr{A}$). The full pipeline $\mathscr{A}$ achieves the best F1 ROUGE-2 performance, showing that the combination of ranking, summarization, and entity extraction is effective for capturing a broader range of n-grams, aligning better with the gold-standard references.

Across the three pipelines $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$, ROUGE-2 scores consistently lag behind baselines in the existing literature, particularly in precision, highlighting potential limitations in capturing all pertinent details despite effectively conveying the main points, as indicated by higher METEOR scores. This suggests opportunities to improve content coverage and lexical alignment. The higher ROUGE-2 recall metrics observed in $\mathscr{C}$, which exclude summarization, may indicate that summarization processes introduced new information, such as the generation of words not present in the original text. This could include the use of synonyms or reformulations, which ROUGE-2 does not account for, as it relies on exact word matching rather than capturing semantic similarities. These observations also raise concerns about the potential loss of critical information during summarization, which should be mitigated in future work.

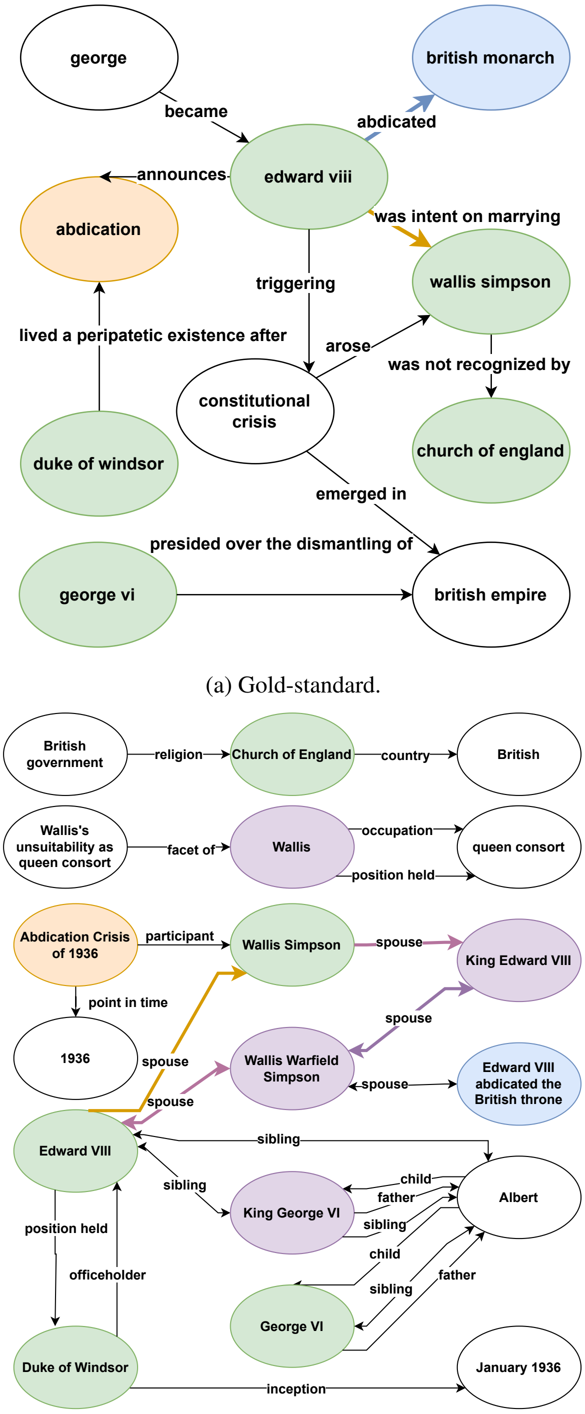
Generally, our higher METEOR and ROUGE-2 recall scores indicate improved summarization quality by emphasizing semantic accuracy and readability over exact word overlap, key factors in our educational context. METEOR, in particular, captures these aspects more effectively than ROUGE-2 (Lin et al., 2022; Schluter, 2017; ShafieiBavani et al., 2018). The lower ROUGE-2 scores compared to Falke et al. may stem from irrelevant or misaligned triples, occasionally resulting in 0.0 scores.

Moreover, our pipeline significantly enhances efficiency with the summarization component, processing each folder in an average of 13s (Wiki-train) and 15s (Wiki-test), compared to 40s and 1 minute with the non-summarization pipeline, a 3-4x speedup, as it can be seen in the processing times logs in the Github.

Qualitative Analysis

Figure 5 shows the gold standard CM from folder 320 of WIKI (Falke, 2019), and the output CM of our full pipeline method. Green and orange colors denote the matching nodes and edges. Green indicates an exact match at the node or edge level, while orange represents semantically similar nodes or edges between the gold standard and our CM. Blue highlights nodes in our CM that are partially similar to the gold standard; for example, the node *"Edward VIII abdicated the British throne"* which is similar to *"(edward viii, abdicted, british monarch)"*. The purple color groups nodes and edges that are semantically similar in our CM. When comparing the gold standard and our CM, we do not find any associations with contradictory meanings.

As shown in Figure 5, our pipeline is capable of generating CMs that are semantically equivalent to the gold standard. However, our performance is affected by non-co-referential resolution. The main concepts are the same or semantically similar, and the only 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 CM, such as the nodes: *"King George VI"* and *"British"*. Furthermore, we notice that our generated CM produces many semantically similar nodes, such as: *"King George VI"* and *"George VI"*, and *"Walls"*, *"Wallis Warfield Simpson"*, and *"Walls Simpson"*. The pipeline's performance could have been enhanced with the capability for co-reference resolution of the concepts.



(a) Gold-standard.

(b) Pipeline generated.

Figure 5: Concepts based on the folder 320 of WIKI TRAIN: gold-standard (left) and generated by $\mathscr{A}$ (right).

The relations between nodes appear to be a more challenging task, with only a small number of corresponding edges. An explanation might be the complex nature of multiple associations between the main concepts in the documents, as the main concepts often have 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.

5.2 LLM-based Methods

Table 6 presents the results of the LLM-based methods compared to the pipeline approaches and the baseline approaches. We observe trends similar to those observed with our pipeline approaches. METEOR scores are higher compared to the ROUGE-2 ones, suggesting that the generated summaries are evaluated more favorably based on linguistic quality metrics rather than exact overlap. Lower ROUGE-2 precision scores suggest that while the generated CM captures crucial information, it faces difficulty in precisely selecting and summarizing essential details without including redundant or unnecessary information.

In line with findings from the literature (Wei et al., 2022), the decomposed prompt outperforms the other two approaches in METEOR scores and ROUGE-2 recall on both the training and the test set. It achieves overall SOTA results on the WIKI TEST dataset, outperforming both pipeline and baseline approaches in METEOR Precision (38.4), F1 score (28.5), and ROUGE-2 Recall (24.3).

6 Conclusion

We propose a neuro symbolic pipeline and a large language model-based method for automated concept map extraction from text evaluated over the WIKI dataset. Our novelty lies in the architecture that utilizes state-of-the-art tools into a neuro-symbolic pipeline with modularized components and its unique application to concept map extraction. Our architecture is the first one to combine symbolic, statistical, and neural technique and to have a summarization step at the beginning of the pipeline. Key technical contributions are the fine-tuned REBEL model and the summarization component, which enhance the originality of the pipeline. Moreover, we analyzed end-to-end LLM-based approaches, which are the first LLM-based end-to-end methods for automated CM extraction. The decomposed prompting method had the best METEOR F1 scores and ROUGE-2 recall, outperforming the current SOTA and effectively competing with supervised and unsupervised methods.

In future work, our aim is to investigate lexical embeddings and semantic rules to increase the performance and accuracy of CM extraction from text. Furthermore, the current metrics used are suitable for text summarization tasks but are not tailored to the CMs generation, as they miss critical aspects of CM creation, such as the graph structure and semantically equivalent concepts, suggesting the need for a new metric. We thus plan to explore evaluation metrics and semantically enhanced benchmarks that are more adapted to this task. In particular, we could explore embedding similarity using a pre-trained language model, or we could also adapt taxonomy metrics such as RaTE and repurposed datasets such as the SemEval 2016 Task 13 (Bordea et al., 2016) to evaluate the quality of concept maps. Moreover, the pipeline should be evaluated on a broader range of texts, encompassing both general and domain-specific content, to assess its robustness across different contexts and to understand how domain knowledge affects performance. This will involve curating more diverse datasets that enable a thorough evaluation and reveal opportunities for further improvement. Additionally, future work should explore post-processing techniques to ensure that key details are preserved in the summarized text, supporting more accurate knowledge representation. A comparative analysis of concept maps generated from both summarized and full-text versions should be done to examine potential trade-offs and better understand the impact of summarization on the overall quality of the concept maps.

7 Limitations

Our methods demonstrate competitive performance compared to baselines from the literature, but also to future areas of improvement. First, the generated CMs reach SOTA performance in the METEOR metric, which demonstrates our pipeline's capabilities. However, the generated CMs might contain lexical variations and paraphrasing, leading to great differences in performance between the METEOR and ROUGE-2 scores (Lavie and Denkowski, 2009; ShafieiBavani et al., 2017). Moreover, reproducing results with OpenAI models can be challenging and inconsistent, even if we used the same summaries from our experiments. To mitigate potential issues such as hallucinations, we consistently set the temperature to 0 when using OpenAI models. Lastly, evaluating beyond quantitative metrics is challenging but crucial for a complete assessment, which is why we conducted an initial qualitative analysis.

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.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1081–1091.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

AJ Cañas and JD Novak. 2006. Jump-starting concept map construction with knowledge extracted from documents.

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

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.

Kuo-Kuang Chu, Chien-I Lee, and Rong-Shi Tsai. 2011. Ontology technology to assist learners' navigation in the concept map learning system. *Expert Systems with Applications*, 38(9):11293–11299.

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.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

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.

Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Martina Galletti, Michael Anslow, Francesca Bianchi, Manuela Calanca, Donatella Tomaiuoli, Remi Van Trijp, Diletta Vedovelli, and Eleonora Pasqua. 2022. Interactive concept-map based summaries for send children.

Ralph E. Gomory. 1958. An algorithm for integer solutions to linear programs.

Mohammad Hadi Hedayati, Mart Laanpere, and Mohammad Arif Ammar. 2017. Collaborative ontology maintenance with concept maps and semantic mediawiki. *International Journal of Information Technology*, 9:251–259.

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.

Kutay Icoz, Mehmet Akif Cakar, Tuncay Yigit, and Samet Egi. 2014. An ontology editor: Creating concept maps for semantic web based e-learning systems. In *INTED2014 Proceedings*, pages 7505–7509. IATED.

Jesper Jensen and Lars Johnsen. 2016. Defining the notion of concept maps 3.0.

Ling Jiang, Zongkai Yang, Qingtang Liu, and Chengling Zhao. 2008. The use of concept maps in educational ontology development for computer networks. In *2008 IEEE International Conference on Granular Computing*, pages 346–349.

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. *Preprint*, arXiv:2210.02406.

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.

Markus Krötzsch, Denny Vrandečić, and Max Völkel. 2006. Semantic mediawiki. In *International semantic web conference*, pages 935–942. Springer.

Alon Lavie and Michael J Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23:105–115.

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.

Wuhang Lin, Shasha Li, Chen Zhang, Bin Ji, Jie Yu, Jun Ma, and Zibo Yi. 2022. Summscore: A comprehensive evaluation metric for summary quality based on cross-encoder. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 69–84. Springer.

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.

Ungkyu Park and Rafael A Calvo. 2008. Automatic concept map scoring framework using the semantic web technologies. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 238–240. IEEE.

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.

Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 41–45. Association for Computational Linguistics.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. 2017. A semantically motivated approach to compute rouge scores. *arXiv preprint arXiv:1710.07441*.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. 2018. A graph-theoretic summary evaluation for rouge. In *EMNLP 2018*, pages 762–762. Association for Computational Linguistics.

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.

Olegs Verhodubs and Janis Grundspenkis. 2013. Algorithm of ontology transformation to concept map for usage in semantic web expert system. *Applied Computer Systems*, 14(1):80–87.

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.