Unlocking Understanding: A Novel Pipeline for Automated Concept Map Extraction from Text

Anonymous ACL submission

Abstract

Concept maps are graphs of entities and their relations that can foster students' understanding of texts. However, manually constructing them is a challenging task. To overcome this, automatic concept map extraction methods have emerged, typically using a pipeline approach to extract entities and their relations. Yet, existing methods face limitations in scalability, scarcity of data and non open-access architectures. To bridge these gaps, we introduce a novel, modularized and open-source pipeline for concept map extraction, using semantic and sub-symbolic techniques. To address scalability, we integrate a summarization step over the input documents and an importance ranking step to make relation extraction more efficient. To tackle data scarcity, we fine-tune a sequence-to-sequence neural model with limited annotated examples. Our approach achieves state-of-the-art performance on METEOR metrics, particularly crucial for concept maps, given the focus on semantic similarity of this metric, and state-of-the-art precision for ROUGE-2. This contribution advances automated concept map extraction, opening doors to wider applications supporting learning and knowledge access.

1 Introduction

Teachers often use concept maps to facilitate students' understanding and meaningful acquisition of information from texts. Novak and Cañas (2007) were the first ones to define concept maps as structured summaries in the form of a graph, as shown in Figure 1, and to observe their capacity to help students in "learning how to learn" (Novak, 1990). Since then, concept maps have been recognized as a useful learning support modality, aiding in integrating new information with pre-existing knowledge (Canas et al., 2001) and potentially benefiting several school or academic disciplines, such as biology or history (Baxendell, 2003).

Manual creation of concept maps from text is however challenging and impractical. 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). Additionally, its potential applications extend beyond learning, as demonstrated by several studies in information retrieval and knowledge representation (Villalon, 2012; Leake, 2006).

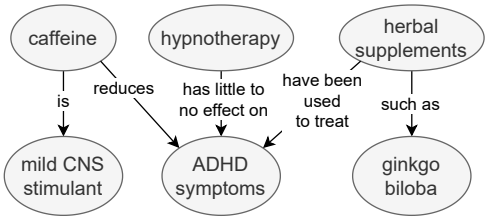

Figure 1: An example for a concept map, showing six concepts and five relations between them. It was created based on a text discussing alternative treatment options for ADHD. Example taken from Falke (2019).

Despite its benefits, the automatic construction of concept maps still exhibit several shortcomings. First, they struggle with scalability, being often limited to processing small document collections and unable to handle large-scale datasets efficiently. Second, they depend on the availability of annotated datasets to implement supervised models efficiently. Third, their code or architecture is not openly accessible. Lastly, these methods do not incorporate external world knowledge, which could significantly enhance the quality of the system generated concept maps.

Our contributions are threefold. First, we introduce a novel and open-access pipeline for automated concept map extraction from text, that leverages semantic and sub-symbolic methods[1]. Second, we demonstrate that fine-tuning relation extraction models with few annotated examples can yield results comparable to supervised models. Such fine-

[1] https://github.com/vs1rr/concept_map_extraction

tuned models help addressing the scalability and limited dataset challenges found in the literature. Lastly, we evaluate our approach and achieve state-of-the-art results on the WIKI dataset. Notably, we emphasize the effectiveness of summarizing the entire document rather than selecting specific portions of the graph, leading to a general enhancement in output quality as evidenced by higher METEOR metrics (34.23, 19.19 and 24.13 for Precision, Recall and F1 respectively), and ROUGE-2 Precision score (21.88).

2 Related Work

Concept maps are structured representations of knowledge (Falke, 2019) and are formally defined as labeled and directed graphs with nodes representing entities and edges representing relations between these entities. Concept map extraction can thus be framed as a summarization task where the summary is in a graph format rather than text.

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 work can be divided into two types of methods: the ones addressing Concept Map - Multi Document Summarization (CM-MDS) and the ones tackling Concept Map - Document Summarization (CM-DS) (Falke et al., 2017). CM-MDS can be seen as a variant of traditional multi-document summarisation approaches that aims to generate one concept map from a text cluster (Falke, 2019). In contrast, CM-DS aims to produce a concept map for an individual document (Falke, 2019).

Early research efforts primarily focused on CM-DS (Oliveira et al., 2001). Subsequent studies employed unsupervised methods with deep syntactic parsing (Leake, 2006) for concept selection, predominantly based on frequency. Going further, Kowata et al. (2010) extracted concept maps from non-English texts, particularly Portuguese news articles. Additionally, de Aguiar et al. (2016) introduced a comprehensive pipeline approach that integrates grammar rules, co-reference resolution, and concept ranking based on occurrence frequency.

Concerning CM-MDS, Rajaraman and Tan (2002) pioneered the domain, utilizing regular expressions and term-frequency-based grouping. Zouaq et al. (2011) later defined patterns over dependency syntax representations for entity extraction and highlighted the utility of concept map mining in ontology learning. Žubrinić et al. (2015) were the first ones to extend the CM-MDS task to languages beyond English, introducing a heuristic approach for summarizing concept maps from legal documents written in Croatian. Table 1 summarizes existing methods for both CM-DS and CM-MDS.

Authors	Language	Task
Oliveira et al. (2001)	English	CM-DS
Rajaraman and Tan (2002)	English	CM-MDS
Leake (2006)	English	CM-DS
Villalon and Calvo (2009)	English	CM-DS
Kowata et al. (2010)	Portuguese	CM-DS
Zouaq et al. (2011)	English	CM-MDS
Zubrinic et al. (2012)	Croatian	CM-MDS
Qasim et al. (2013)	English	CM-MDS
Žubrinić et al. (2015)	Croatian	CM-MDS
de Aguiar et al. (2016)	English	CM-DS
Falke and Gurevych (2017)	English	CM-MDS
Falke et al. (2017)	English	CM-MDS
Falke (2019)	English, German	CM-MDS
Yang et al. (2020)	English	CM-MDS
Nugumanova et al. (2021)	Kazakh, Russian	CM-MDS
Ghodratnama et al. (2023)	English	CM-MDS
Lu et al. (2023)	English	CM-MDS
Bayrak and Dal (2024)	Turkish	CM-MDS

Table 1: Pipeline methods for CM-DS and CM-MDS, with the languages in which the model is available.

Lastly, it is noteworthy to mention the work of Falke (2017; 2017; 2019), who made significant contributions to the field and whose datasets act as the main benchmark for evaluating our approach. They formalized the task and definitions of automated concept map extraction, introduced new shared evaluation protocols, created benchmark datasets, and developed new model pipelines. Their model leverages predicate-argument structures and automatic models for German and English, achieving state-of-the-art performance in both CM-DS and CM-MDS. The model presented by Falke et al. (2017) is a pipeline approach including four distinct steps: (1) Concept and Relation extraction, relying on Open Information Extraction with filtering and sub-processing; (2) Concept Graph construction with a system based on pairwise classification and set partitioning; (3) Graph summarization using Integer Linear Programming.

Inspired by past and current approaches, we maintain a pipeline architecture, while introducing an open access and modularized system which is based on semantic and sub-symbolic methods. Furthermore, we fine-tune a sequence-to-sequence model based on BART (Lewis et al., 2019) for the

relation extraction sub-task, overcoming the limited scalability of current models and limited access to resources. Moreover, we introduce preliminary summarization and importance ranking steps that reduce the search space and enhance the scalability of the pipeline.

3 Model Pipeline

Our model consists in a pipeline with four components, as depicted in Figure 2: (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. In this section, we further describe each component of our pipeline.

3.1 Summary Extraction

We integrate methods for extractive and abstractive summarization. Extractive summarization involves selecting and arranging key sentences or phrases directly from the source text, while abstractive summarization produces a concise summary of a document by paraphrasing and generating new sentences to capture the essence of the original text (Mahajani et al., 2019).

For extractive summarization, we implemented LexRank (Erkan and Radev, 2004) with the Sumy package[2]. LexRank is a graph-based approach that uses centrality measures and that is computationally efficient. We chose this method as it was used in previous work for concept-based extractive summarization (Chitrakala et al., 2018). For abstractive summarization, we use the transformer-based model *gpt-3.5-turbo-0125*[3] 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 the summary.

3.2 Importance Ranking

Importance ranking identifies the most salient sentences within a document or a cluster of documents. We implemented three methods in our pipeline. Two of them rely on word embeddings and statistical measures, while the third one is a graph-based technique.

The first technique is based on Word2Vec (Mikolov et al., 2013), that was accessed through the Gensim library[4]. After training the Word2Vec model, each sentence in the document is transformed into a vector representation by computing the average embedding of its words. The importance score of each sentence is then the sum of its cosine similarity with every other sentence in the document. We chose Word2Vec as we wanted to mimic recent work on neural extractive summarization which involved sentence extraction based on embeddings (Cheng and Lapata, 2016). Cosine similarity was used due to its capacity to capture semantic similarity between two vectors : sentences with similar meanings will have a higher cosine similarity scores, thus sentences more semantically related to each other are prioritized in the ranking phase.

The second one is based on Term Frequency-Inverse Document Frequency (Christian et al., 2016), or Tf-Idf. We computed the representation of each sentence in the original document using Scikit-learn[5]. The importance ranking score of a sentence is obtained by summing the values of its Tf-Idf representation. We chose Tf-Idf as we wanted to assess the performance of the Word2Vec-based method over a traditional statistical approach. In particular, we wanted to assess the importance of word-frequencies or document-specific attributes in determining the ranking of sentences, in comparison to their embedding and semantic similarity.

The third one is PageRank (Page et al., 1999), implemented using the NetworkX package[6] which was selected due to its establishment as a baseline in prior research (Falke et al., 2017). PageRank was originally computed from a graph representation of the web pages, with pages as nodes and hyperlinks as edges, 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. For this method, the sentences of each document were converted into a matrix of token counts.

The importance ranking step can be done on each document individually ($single$), or for all documents combined (all). Similarly to summarization (Section 3.1), we also add as parameter a $ranking\_perc\_threshold$.

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

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

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

[5] https://scikit-learn.org/stable/

[6] https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html

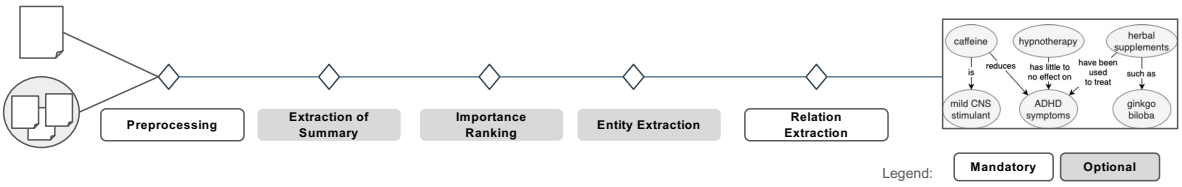

Figure 2: Outline of our method from input to output. The output's concept map is the same as Figure 1.

3.3 Entity Extraction

Entity extraction is used to extract relevant entities from text. We used DBpedia Spotlight[7] that is easily accessible and has high accuracy (Mendes et al., 2011). We add the $confidence$ as parameter in our model. For example, a confidence of 0.5 indicates that only the entities with a confidence level higher than 0.5 are retrieved.

3.4 Relation Extraction

The task of relation extraction plays a crucial role in CM-MDS and CM-DS as it identifies relations between entities. As in Huguet Cabot and Navigli (2021), we refer to (triple) relation extraction as the task of extracting triples from raw text in the form $(subject, predicate, object)$, with no given entity spans. For this sub-component we fine-tuned REBEL (Huguet Cabot and Navigli, 2021), a sequence-to-sequence model based on BART (Lewis et al., 2019) which was mostly trained on Wikipedia abstracts and contains over 200 relation types. Consequently, the model was trained on a fixed set of relations, whereas the number of relations in concept maps are not constrained (Falke, 2019). We chose REBEL because it achieved state-of-the-art performance across multiple tasks and it contains a reasonable number of parameters compared to other systems such as UniREl (Tang et al., 2022) or DEEPSTRUCT (Wang et al., 2022). It is furthermore openly available and can be easily fine-tuned to extract new relations.

We fine-tuned REBEL using relations extracted from BIOLOGY (Olney et al., 2011) to assess its performance on concept map relation extraction. 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). The final corpus resulted in 183 English document-concept-map pairs. One example can be found in Table 2.

[7] https://github.com/dbpedia-spotlight/dbpedia-spotlight

Reference Concept Map
(allele, can be used to predict, genotype)
(diploid organism, have, one copy of each allele)
(allele, is alternative form of, gene)

Table 2: Example of reference concepts maps in BIOLOGY on the topic of alleles.

However, the BIOLOGY corpus maps one document with several sentences to one concept map. In contrast, relation extraction operates at the individual sentence level. Consequently, we found it necessary to perform additional post-processing to prepare the data for training. With that goal in mind, we developed a rule-based system that takes a sentence and a triple as input, returning a boolean value indicating whether the information in the triple is present in the sentence. We checked the presence of both the subject and the object in the sentence, as well as the lemmas of the predicate's verbs. This process resulted in 220 mappings which we subsequently divided into training, evaluation, and test sets for fine-tuning. The split for train/eval/test was 80/10/10. The fine-tuning was conducted using the following parameters: $learning\_rate = 2.5 * 10^{-5}$, $epochs = 10$, $batch\_size = 4$, $seed = 1$. In our experiments, we compare the base REBEL to our fine-tuned REBEL.

4 Experimental Setup

4.1 Data and Baselines description

We used the WIKI dataset (Falke, 2019) as a benchmark for CM-MDS. Examples of Concept Maps taken from this dataset can be found in Table 3.

WIKI was obtained through an automated corpus extension method that integrates automatic pre-processing, scalable crowd-sourcing, and high-quality expert annotations (Falke, 2019). It comprises 38 clusters of English documents, each centered on a distinct topic, and divided in half for train and test set. Each cluster comes with a ref-

Reference Concept Map
(7 world trade center, is, a building in new york city)
(new york city, located across, from world trade center site)
(world trade center, was, a building in new york city)

Table 3: Example of a reference standard concept map taken from the WIKI corpus. It was extracted from a cluster of documents about the World Trade Center.

erence concept map. This dataset 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 evaluated our model on WIKI only for the CM-MDS task. We did not test our model for the CM-DS task on the BIOLOGY corpus, since it was used to fine-tune the REBEL model.

We added a pre-processing step that transformed all text to lowercase, removed punctuation, eliminated double spaces, and filtered out noise-prone information such as contributors, web links and bibliography references.

We compare our model against supervised and unsupervised methods proposed in previous studies, following the approach of Falke et al. (2017). Specifically, for unsupervised methods, we compare to PageRank (Page et al., 1999), Leake (2006), Žubrinić et al. (2015). For supervised models, we present results from Falke and Gurevych (2017), and Falke et al. (2017).

4.2 Evaluation Metrics

Falke (2019) introduced two automatic metrics based on METEOR and ROUGE. METEOR (Banerjee and Lavie, 2005) is a metric used to evaluate machine translation quality, comparing candidate and reference translations based on word or phrase alignment which accounts for synonyms and paraphrases [8]. ROUGE (Barbella and Tortora, 2022) is a metric used to evaluate the quality of summaries by comparing them to reference summaries. It measures the overlap of n-grams (sequences of n words) and their variants between the system-generated summary and the reference summaries[9].

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 $\mathbf{s_r}$. The propositions were separated with ".", to avoid counting overlaps across triples. 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.

4.3 Constant parameters

In this section, we provide details on the constant parameters used for our experiments, while the variable parameters are described in Section 4. We split and ran our experiments on two Ubuntu machines with 2 GPUs, 40 CPUs, and 348GiB of memory.

For the summarization components (Section 3.1), we focused solely on document-level summarization, instead of cluster-level summarization. This decision stemmed primarily from limitations on the prompt size in the OpenAI API: concatenating all texts would have exceeded the prompt's capacity. When using *gpt3.5-turbo-0125*, we set a $temperature$ of 0, to keep the summary as close to the original text as possible. To avoid repeatedly running the summarization process with OpenAI, we first stored them and re-used them as cache during the experiments. It cost 2$ in total.

For more efficiency in the entity extraction phase (Section 3.3), we set up a local DBpedia Spotlight API, following instruction from *spacy-dbpedia-spotlight*[10]. We used $en\_core\_web\_lg$ for the spaCy model.

For relation extraction with REBEL-based or REBEL-derived models (Section 3.4), we used the openly available REBEL tokenizer *Babelscape/rebel-large*[11].

[8] For this work, we used METEOR 1.5

[9] For this work, we used ROUGE 1.5.5

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

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

5 Results

5.1 Hyperparameter tuning

We first experimented on the training dataset of WIKI to select the most meaningful parameters for some of the components. Table 4 shows the different parameters that were tested. For the summary part, we investigated the impact of $summary\_method$ and $summary\_percentage$ on the quality of the concept maps. For the ranking part, we looked at the different methods ($ranking$), as well as the impact of individual vs. multi-document level ranking ($ranking\_how$) and various percentage thresholds ($ranking\_perc\_threshold$). For entity, we experimented with different $confidence$ scores, and with the two $relation$ models for relation extraction.

The different sets of parameters resulted in 768 different parameter combinations. We hereafter describe the main results, and make the full results with metrics available together with our code[12].

Table 4: List of all parameter values for each component. Ranking level $single$ with a percentage threshold of 15 means that for each document, the top 15% sentences were retained for entity and relation extraction. Ranking level all with a percentage threshold of 15 means that the top 15% sentences with all documents together were retained for entity and relation extraction. $rebel\_hf$ refers to the base REBEL model, while $rebel\_ft$, the fine-tuned REBEL model.

Component	Parameters	Values
Summary	$summary\_method$	*gpt3.5-turbo-0125* | *LexRank*
	$summary\_percentage$	15 | 30 | 50 | 70
Ranking	$ranking$	$word2vec$ | $page\_rank$ | $tfidf$
	$ranking\_how$	$single$ | all
	$ranking\_perc\_threshold$	15 | 30 | 50 | 70
Entity	$confidence$	0.5 | 0.7
Relation Extraction	$relation$	$rebel\_hf$ | $rebel\_ft$

We first looked at the correlation between parameters for which only two options were possible from Table 4 - later called binary parameters - and the macro-averaged F1 scores for METEOR and ROUGE-2. The parameters that are concerned are: $summary\_method$, $ranking$, $confidence$ and $relation$.

The results can be found in Table 5. We found a moderate to strong positive correlation between the following parameters and both METEOR and ROUGE-2 F1 scores: *gpt3.5-turbo-0125* for summarisation ($corr = 0.53$ and $corr = 0.62$ for METEOR F1 and ROUGE-2 F1 respectively), ranking sentences for all sentences in documents rather than at document-level ($corr = 0.31$ and $corr = 0.03$), and using the fine-tuned REBEL model for relation extraction ($corr = 0.15$ and $corr = 0.40$). All the correlations were statistically significant, with $pval < 0.05$, and usually $pval << 0.05$. However, there was no meaningful impact of the confidence score for the entity extraction. We believe that the confidence level was high for all entities, making the distinction between 0.5 and 0.7 not statistically significant.

We then further looked at the three ranking options: $page\_rank$, $word2vec$ and $tfidf$. We found that there was no stronger options with respect to the ROUGE-2 F1 scores. However, when comparing each ranking individually with the two others (eg., $page\_rank$ vs. $word2vec$ & $tfidf$) with respect to the METEOR F1 scores, we found that $page\_rank$ had a moderate positive correlation ($corr = 0.21$ and $pval = 5.0 \times 10^{-9}$), and that $tfidf$ had a moderate negative correlation ($corr = -0.27$ and $pval = 2.6 \times 10^{-14}$).

We lastly compared the parameters $summary\_percentage$ and $ranking\_perc\_threshold$ across the experiments with parameters that correlated strongly with better F1 scores: $summary\_method$ = *gpt3.5-turbo-0125*, $ranking = page\_rank$, $ranking\_how = all$, $confidence = 0.5$, $relation = rebel\_ft$. Although the METEOR scores were quite similar, the difference was much more noticeable for the ROUGE-2 scores. Overall, we found that lower values of $summary\_percentage$ and $ranking\_perc\_threshold$ yielded better results, and that the best scores were achieved for $summary\_percentage = 15$ and $ranking\_perc\_threshold = 15$.

For computational and performance reasons, we therefore kept the following parameters to evaluate our pipeline on the test set: $summary\_method =$ *gpt3.5-turbo-0125*, $summary\_percentage = 15$, $ranking = page\_rank$, $ranking\_how = all$, $ranking\_perc\_threshold = 15$, and $relation = rebel\_ft$.

5.2 Evaluation Results

All the results for the training and test sets of WIKI are presented in Table 6, where our results are also compared to the baselines detailed in section 4.1. Table 6 displays results for several variations of

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

Table 5: Correlation between binary features and F1 scores. The table reads as follows: a correlation of 0.533 for meteor_f1 means that there is a positive correlation between *gpt3.5-turbo-0125* summarisation and meteor_f1, compared to $LexRank$ summarisation.

Feature	Value 1	Value 2	Metric	Correlation	P-value
$summary\_method$	*gpt3.5-turbo-0125*	$LexRank$	meteor_f1	0.533	$1.69e-57$
			rouge-2_f1	0.622	$2.79e-83$
$ranking\_how$	all	$single$	meteor_f1	0.308	$2.38e-18$
			rouge-2_f1	0.035	$3.39e-01$
$confidence$	0.5	0.7	meteor_f1	0.003	$9.36e-01$
			rouge-2_f1	0.0	$9.93e-01$
$relation$	$rebel\_ft$	$rebel\_hf$	meteor_f1	0.154	$1.79e-05$
			rouge-2_f1	0.398	$1.43e-30$

component combinations : (A) Summary, Importance Ranking, Entity Extraction, Relation Extraction (B) Summary Extraction, Entity Extraction, Relation Extraction (C) Importance Ranking, Entity Extraction, Relation Extraction. Our top performing system (B) achieved state-of-the-art METEOR Precision and F1 scores of 34.23 and 24.13, and a state-of-the-art ROUGE-2 Precision score of 22.10. However, the results for ROUGE-2 metrics for (B) fell below existing baselines, with Recall and F1 values of 2.65 and 4.56 respectively. (A) achieved the state-of-the-art METEOR Recall score of 23.37.

(A) demonstrates superior semantic similarity, coherence and consistency compared to bi-gram overlap only. However, while the system concept map seems to effectively convey the main points (higher METEOR score), it may lack in capturing all relevant details (lower ROUGE-2 score), suggesting potential for improved content coverage and lexical alignment. This could be due to potential inaccuracies in the summarization process, which may omit or incorrectly summarize text.

(B) and (C) achieved a high precision and low recall for the ROUGE-2 metric, and competitive METEOR results. This suggests that the system is good at correctly identifying relevant instances (high precision) but may miss many of the relevant instances present in the data (low recall). While looking at the differences between examples, we noticed that combining too strong percentages for summary extraction and importance ranking could impact negatively the quantity of triples extracted.

For an initial qualitative evaluation, we refer to Figures 3 and 4 that show examples of system generated concept maps vs. the reference concept map, for a good example and a bad example respectively. For more clarity, we shortened the two concept maps, but we make them available together with the code [13]. In the good example (Figure 3), one can see that the entities mostly overlap, like *Anish Kapoor* or *Cloud Gate*. The main difference is on the non-named entities which are in the reference concept map, such as *public* or *park grill* and not in the system-generated concept map, since DBpedia contains mainly named entities. Furthermore,in the system-generated, there are some redundant triples concept map, and two unrelated triples concerning *bean* and *killer whale*. In the bad example (Figure 4), one can see that the entities are very different and that the two concept maps could be on different topics. After looking more closely through the different steps of our system, we found that the main error comes from the OpenAI model that does not produce a correct summary, therefore leading to the extraction of different entities and relations.

6 Limits & Risks Statement

Our pipeline architecture demonstrate competitive capabilities compared to baselines, yet it has some limitation. First, the pipeline architecture could inherently lead to error propagation, potentially impacting downstream tasks' results. This is the case for the trade-off of summary extraction and importance ranking percentages. Second, the triples extracted seems to be of good quality but the observed discrepancies indicated by lower ROUGE-2 scores suggest a possible omission of triples in our system's output. After manual inspections, we also found that some extracted triples are redundant and some duplication step should be added in our relation extraction model. Third, reproducing results with the OpenAI models can sometimes be

[13]The figures are available at this link.

Table 6: Results for all systems on WIKI TRAIN and TEST. *RE: refers to the Relation Extraction(only mandatory component in our pipeline)*; *E: refers to the Entity extraction*; *"-" indicates that we couldn't access to the results.*

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) Summary + Ranking + E + RE	24.36	28.32	25.49	12.8	11.86	11.39	21.89	**23.37**	21.88	9.77	8.11	7.71
(B) Summary + E + RE	34.44	21.43	25.95	23.63	3.81	6.36	**34.23**	19.19	**24.13**	**22.10**	2.65	4.56
(C) Ranking + E + RE	34.27	19.68	24.73	22.59	2.86	4.97	33.24	16.56	21.9	20.84	1.82	3.31

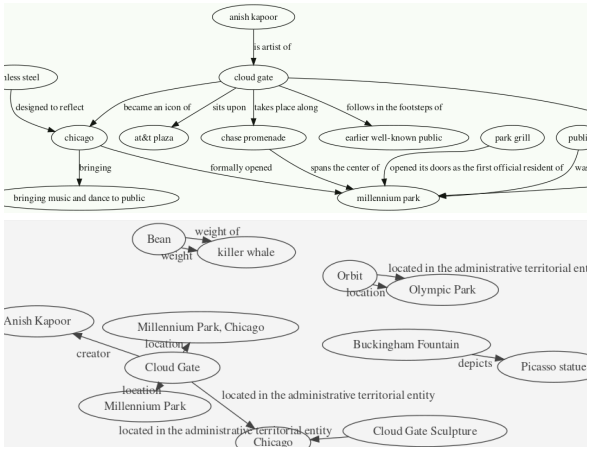

Figure 3: Reference (upper) and system generated (lower) concept maps for folder 133 of the WIKI test corpus. For METEOR, $Pr = 32.46$, $Re = 33.63$ and $F1 = 33.03$. For ROUGE-2, $Pr = 14.13$, $Re = 19.70$ and $F1 = 16.46$. The cluster of documents was around the Millennium Park in Chicago and particularly about one of its sculpture, Cloud Gate.

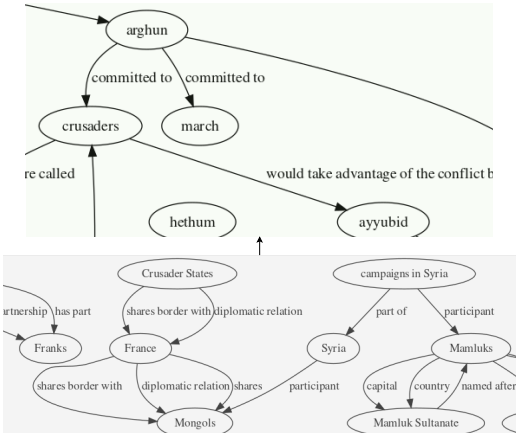

Figure 4: Reference (upper) and system generated (lower) concept maps for folder 241 of the WIKI test corpus. For METEOR, $Pr = 14.48$, $Re = 15.88$ and $F1 = 15.15$. For ROUGE-2, $Pr = 2.38$, $Re = 1.23$ and $F1 = 1.62$. The cluster of documents was around the expansion of the Mongol Empire.

challenging and inconsistent, even if we submit the summaries we worked with in our experiments. Lastly, evaluating beyond quantitative metrics can also be challenging.

Lastly, we acknowledge some risks associated to the improper use of our pipeline, such as the summaries containing hallucinations, which could therefore convey fake or irrelevant information. To mitigate these risks, we plan to make our pipeline available only for research purposes.

7 Conclusion & Future Work

In this paper, 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-tuned a sequence-to-sequence model for relation extraction, and used external knowledge bases for entity extraction. We evaluated our method against an annotated dataset for Concept Map based Multi-Document Summarization. The method that achieves the best results is the one combining all components except importance ranking. This approach achieves state-of-the-art performance for METEOR metrics and for ROUGE-2 Precision, competing with both supervised and unsupervised methods.

In future work, we aim to improve the relation extraction part. One direction is to implement a redundancy control after relation extraction, and to integrate more fine-grained relation extraction. We also aim to investigate further low recall scores, and to integrate qualitative metrics for future evaluation.

References

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.

Marcello Barbella and Genoveffa Tortora. 2022. Rouge metric evaluation for text summarization techniques. *Available at SSRN 4120317*.

Brad W Baxendell. 2003. Consistent, coherent, creative: The 3 c's of graphic organizers. *Teaching Exceptional Children*, 35(3):46–55.

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.

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.

Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. 2016. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294.

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

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.

Samira Ghodratnama, Amin Behehsti, and Mehrdad Zakershahrak. 2023. A personalized reinforcement learning summarization service for learning structure from unstructured data. In *2023 IEEE International Conference on Web Services (ICWS)*, pages 206–213. IEEE.

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.

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*.

Jiaying Lu, Xiangjue Dong, and Carl Yang. 2023. Weakly supervised concept map generation through task-guided graph translation. *IEEE Transactions on Knowledge and Data Engineering*.

Abhishek Mahajani, Vinay Pandya, Isaac Maria, and Deepak Sharma. 2019. A comprehensive survey on extractive and abstractive techniques for text summarization. *Ambient Communications and Computer Systems: RACCCS-2018*, pages 339–351.

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*.

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 Alberto J Cañas. 2007. Theoretical origins of concept maps, how to construct them, and uses in education. *Reflecting education*, 3(1):29–42.

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.

Jorge Villalon and Rafael A Calvo. 2009. Concept extraction from student essays, towards concept map mining. In *2009 Ninth IEEE International Conference on Advanced Learning Technologies*, pages 221–225. IEEE.

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*.

Carl Yang, Jieyu Zhang, Haonan Wang, Bangzheng Li, and Jiawei Han. 2020. Neural concept map generation for effective document classification with interpretable structured summarization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1629–1632.

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.