An AMR-based Extractive Summarization Method for Cohesive Summaries

No Author Given

No Institute Given

Abstract. The Extractive Summarization approach, widely employed in automatic text summarization, aims to identify and extract the most informative sentences from a document and generate a summary. This paper presents a novel method for mono-document extractive summarization that leverages Abstract Meaning Representation (AMR) as a semantic representation of each sentence. In our method, an AMR parser is employed to capture the core concepts of each sentence, expressed in core semantic terms. To enhance both the relevance and cohesiveness of the summary, a concept-based Integer Linear Programming (ILP) technique is employed. We conducted experiments on two DUC datasets (2001/2002) which show that our proposed system achieved either higher or competitive performance against other state-of-the-art summary systems.

Keywords: AMR · Summarization · Extractive Summarization · Cohesive Summaries

1 Introduction

The demand for concise information centered around key concepts has driven extensive research in the field of Automatic Text Summarization (ATS) [1]. According to Tas and Kiyani [2], the primary goal of ATS is to condense the original text into a shorter version, while retaining the information content and general meaning.

ATS can be broadly categorized into two types: extractive and abstractive methods. The extractive summarization method selects key sentences from a document based on their statistical significance, considering factors like word frequency and sentence position. By assigning weights to words and sentences, the most informative ones are chosen for composing final summary [3]. In contrast, the abstractive summarization approach generates summaries that go beyond the extraction of sentences from the original text. It leverages various techniques from natural language processing (NLP), including semantic analysis, language generation, and paraphrasing, to produce summaries that may contain rephrased or synthesized content not present in the original text [4].

ATS can also be categorized according to the number of documents analyzed simultaneously. The mono-document approach produces a final summary based on a single document, while the multi-document approach processes more than one document to generate a single summary. Due to the complexity of implementing the abstractive approach, the extractive approach is commonly employed and will be the primary focus of this paper.

Ensuring the quality of automatically generated summaries poses a challenge in the field of ATS. An effective summary should encompass the most pertinent information while excluding redundancies, ensuring consistency, and facilitating comprehension [4]. One limitation of existing ATS systems is their tendency to prioritize informativeness over cohesiveness. Coherence and logical connectivity between sentences are vital for promoting reader understanding. Hence, human evaluation becomes necessary to assess both the informational content and the cohesion of a summary. To mitigate this issue, we propose leveraging semantics by employing a semantic representation of document sentences as a means to enhance the relevance and cohesion of extractive summaries.

This paper proposes a novel method to enhance the extractive summarization process in terms of informativess and cohesion by leveraging Abstract Meaning Representation (AMR) [5]: as a semantic representation for the sentences in a document. Our framework for semantic representation is able to capture the underlying semantic meaning of sentences. In the proposed solution, we employ a concept-based Integer Linear Programming (ILP) [26] technique to extract the most relevant concepts from sentences in a document. By considering the underlying concepts and their relationships, our system ensures that the selected sentences are not only individually relevant but also collectively cohesive, resulting in a more coherent and informative summary. To evaluate the effectiveness of our system, we conducted experiments on two DUC datasets (2001/2002) [1] which show that our proposed system achieved either higher or competitive performance against other state-of-the-art summary systems.

This paper is organized as follows. In Section 2, we review related work on automatic summarization using semantic representations. Section 3 presents the foundations of AMR semantic representation. Section 4 describes our extractive summarization method based on AMR. In Section 5, we discuss the experimental results of our summarization system on two benchmarking corpora, comparing it with other extractive summarization systems.

2 Related Work

The role of semantic representations for summarization has been explored in several works. For instance, [14] proposed a method based on AMR to generate an abstract summary. This approach involves three-step process: (1) using an AMR parser to generate sentence graphs for each document, (2) merging and transforming these individual AMR graphs into a single AMR graph, and (3) generating a summary based on the information encapsulated within the summary graph. Another work by Dohare et al. [7] proposed an abstractive summarization system incorporating AMR as an intermediate step. The system initially generates the AMR graph for a given input document, extracts an

[1] `https://www-nlpir.nist.gov/projects/duc/index.html`

abstract graph from it, and finally generates summary sentences based on this abstract graph. Although these approaches are interesting, they can be computationally intensive and challenging to implement for larger documents and sentences.

While extractive summarization remains the predominant approach in automatic summarization, researchers have explored ways to enhance its effectiveness by incorporating sentence semantics. One notable study by [17] proposes a concept-based ILP approach for single-document summarization. Their unsupervised method is based on ILP and introduces a novel weighting technique that combines sentence coverage and position to estimate concept importance. Additionally, a weighted distribution strategy is proposed, prioritizing sentences at the beginning of the document when they contain relevant concepts. Their system was evaluated on three datasets, including the DUC 2001-2002 and the CNN corpus.

H. Oliveira et al. [18] propose a regression-based approach integrating ILP for single-document summarization. Their approach employs a concept-based ILP method to generate multiple candidate summaries for each input document. A regression model is then applied with several resources extracted from the summary, sentences, and n-gram levels to select the most informative summary from the generated candidates. The effectiveness of their approach is evaluated on well-known datasets, including DUC 2001-2002, and the CNN corpus. Although the aforementioned works focusing on the extractive approach offer valuable insights, they might not fully capturer the semantics of the sentences in the document. Incorporating a true semantic representation would enable a more comprehensive understanding of the underlying meaning, leading to potential enhancements in the summarization process by considering the semantic nuances of the text.

3 Abstract Meaning Representation

Abstract Meaning Representation (AMR) consists in a widely used formalism for the semantic representation of sentences. It identifies the underlying meaning of the text, mainly the agents, actions, and recipients involved. AMR encompasses semantic relationships, named entities, co-reference, denial, modality, and more. Its main benefit lies in characterizing the semantics of a sentence in an abstract representation, removing syntactic elements such as articles, numbers, verb tense, and voice. AMR, as described in [5] and [6], captures the predicate-argument structure of a sentence using external semantic resources such as lexicons. In AMR, verb semantics is based on the PropBank annotation scheme [21].

In AMR notation, a sentence meaning is captured by merging different grammatical realizations into a unified representation. This representation can be expressed in three formats: first-order logic (FOL), directed acyclic graph (DAG) with a single root, and a textual format inspired by Penman's notation [25]. .

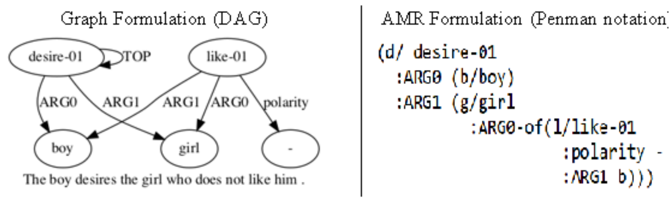

Fig. 1: AMR formulations of the sentence "The boy desires the girl who does not like him".

Figure 1 shows AMR representations of the sentence "The boy desires the girl who does not like him" in both DAG and Penman notation formats.

AMR Parsing Cai et al. introduce a top-down AMR parser model called Graph Spanning based Parsing (GSP) [7]. This parser, also referred to as the "Core Semantic First AMR parser," prioritizes the most significant words or concepts in the sentence closer to the root of the generated graph. According to their findings [7], the Core Semantic First AMR parser produces more precise and hierarchical AMR graphs than other existing AMR parsers.

The GPS AMR parser proves to be particularly valuable for extractive summarization tasks due to its ability to position the most significant concepts near the root of the AMR DAG. This placement facilitates the extraction of key concepts from each sentence in a document by traversing the AMR graph from the root downwards until a desired level is reached. Given the significance of capturing the "core semantic" of a sentence in extractive summarization, our primary hypothesis aligns with the aforementioned properties of the GPS AMR parser.

4 An AMR-based Extractive Summarization Method

This section introduces our AMR-based Extractive Summarization method, outlining its main process comprising multiple steps. We proceed by providing a detailed explanation of each of these steps.

4.1 Main Extractive Summarizatiton Process

Our AMR-based Extractive Summarization method is composed of multiple steps, as illustrated in the pipeline in Figure 2. The first step entails AMR parsing, which involves parsing the sentences of the input document to be summarized into AMR graphs. Next, in the concept extraction step, the core semantics of each sentence are captured by extracting a set of concepts from the corresponding AMR graph. The third step involves mapping sentences to concepts, where relations between sentences are established based on the number of common concepts, resulting in a sentences-concepts graph. In the fourth step,

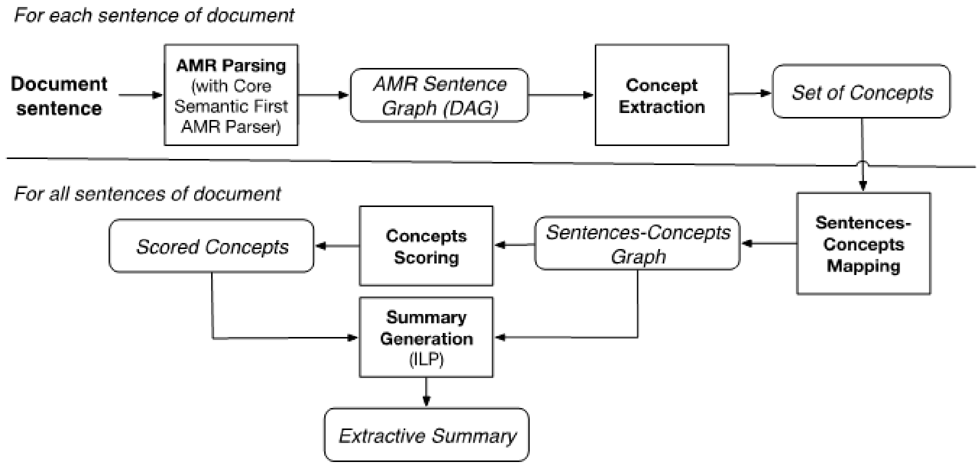

Fig. 2: Main process of the proposed method.

concept scoring is performed, assigning scores to each extracted concept based on its importance within the text, enabling the ranking of sentences containing the most informative concepts. Finally, in the generation step, an ILP-based technique is employed, leveraging the scored concepts and the relational sentence graph to generate the extractive summary. This ILP technique selects the minimum subset of sentences to maximize the coverage of important concepts. The remainder of this section discusses each step in further detail.

4.2 AMR Parsing

In this step, we employ the Core Semantic First AMR Parser introduced by Cai et al. [7]. As mentioned earlier, this parser assigns higher significance to the concepts closer to the root of the AMR DAG. This facilitates the extraction of key concepts from each sentence in the document. By traversing the AMR graph from the root towards the leaves until a given level, the core concepts can be identified and extracted.

4.3 Concept Extraction

The goal of this step is to extract concepts from the sentences represented in the AMR format. For that, we developed an algorithm for parsing the AMR graph in Penman notation. In the AMR graphs, nodes correspond to concepts, while edges represent relationships between these concepts [20]. However, not all relationships are relevant for the summarization task. We have created a list of "Stop-Edges" consisting of 44 relations that we disregard during the extraction process, including date entities such as (:day, :month, :year), quantity-related concepts such as (:quant, :unit, :scale), and certain semantic relations such as (:consist-of, :age, etc.).

To align the concepts in the AMR graph with the original words in the sentence, we propose a rule-based algorithm. Besides the fact that this algorithm generally performs effectively, certain situations require special treatment. For example, let us examine the concept "appropriate" in Fig. 3, which has a relationship with ":polarity -". This denotes that the original word is the negation of the concept "appropriate," i.e., its antonym, "inappropriate." Table 1 shows the core extracted concepts for two sentences (S1 and S2) using our rule-based algorithm.

```
(c/comment
  : (a / appropriate
       :polarity -))
```

Fig. 3: Penman Formulation of the AMR representation for the sentence "The comment is inappropriate" (Banarescu, 2018)

Table 1: Sentences and Selected Concepts

ID	Sentence	Selected Concepts
S1	Congressmen to Sue Census Over Count of Illegal Aliens	Sue Census Count Congressmen Illegal
S2	A coalition of Legislators announced Wednesday that they plan to sue the Census Bureau in an effort to force the agency to delete illegal aliens from its count in 1990	plan coalition sue Census Bureau effort Legislators illegal

Table 2: Selected Concept (Vocabulary)

ID	Concept
C1	Sue
C2	Census
C3	Count
C4	Congressmen
C5	Illegal
C6	Plan
C7	Coalition
C8	Bureau
C9	Effort
C10	Legislators

4.4 Sentences-Concepts Mapping

In our solution, the relationship between sentences is determined by the shared concepts, captured in the sentence graph. This graph denotes sentences and concepts as nodes, with edges directed from sentence nodes to the concepts they contain, as depicted in Fig. 4. To reinforce the connection between sentences, we build a similarity matrix between a given pair of words in the document's vocabulary. This allows words with higher similarity belonging to distinct sentences, to be shared by two sentences. Table 2 shows the set of concepts extracted from sentences S1 and S2, which form the vocabulary of the document comprising these sentences.

Next, we use the vocabulary constructed from the extracted concepts to build the sentences-concepts graph, as depicted in Fig. 4 (a). This graph captures the relationships between the sentences and the concepts in the document. To enhance this graph, we apply a similarity measure to identify similar concepts and add new edges between them. For instance, in the provided example from Tables 1 and 2, we recognize that the words "Congressmen" in sentence 1 and "Legislators" in sentence 2 are similar and connect them with an edge in the updated graph shown in Fig. 4 (6.b). By including such edges, we increase the number of common concepts between the sentences, thereby enhancing relationships between sentences. In this example, initially, there were only three common concepts between the sentences. However, after applying our similarity-based approach, the relationship between the sentences improved, and now there are five common concepts shared between them.

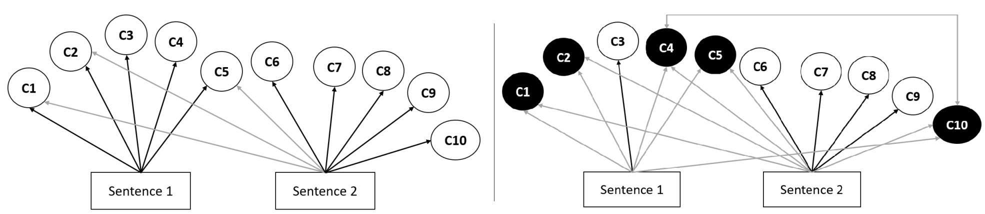

Fig. 4: Graphs of the sentences S1 and S2. (a) Original concepts shared between the sentences.; (b) additional concepts shared between the sentences.

4.5 Concept Scoring

To rank the importance of each concept and determine the sentences containing the most relevant concepts, we assign a score to each concept. Our approach incorporates four concept scoring metrics: Word Frequency, TF-ISF, Lv-Pos, and S-Pos. The Word Frequency and TF-ISF [?] metrics rely on concept frequency within the entire document. The Lv-Pos and S-Pos metrics consider the concept level within the AMR graph. By combining these four metrics, we can generate a comprehensive ranking of the concepts, enabling us to identify the most relevant ones for the summarization task.

Level-based position (Lv-Pos) Lv-Pos is a concept scoring metric that evaluates the importance of a concept within a sentence based on its position in the AMR hierarchy. As the AMR parser follows a top-down structure, where the concepts closer to the root of the hierarchy are generally considered more important [5]. Therefore, the lower the level of a concept in the AMR hierarchy, the more relevant it is considered by our system.

$$Lv - pos(w, si) = 1 - \frac{lv(w, si)}{L} \tag{1}$$

- lv (w, si), returns the level of the word w within a sentence in the AMR graph;
- L, is the maximum level of a sentence in the AMR Graph.

Sum of Level Position (S-pos) The S-pos metric is a document-level score that evaluates the level of concepts in the AMR across all sentences. It is calculated as a normalized sum of all L-scored concepts. This metric takes into acount that the most important concept is the one that appears most frequently and is closer to the root in the AMR graph. Therefore, concepts with a higher S-pos score are deemed more relevant to the overall meaning of the document.

$$S - pos(w) = \frac{\sum_{i=0}^{n} Lv - pos(w, si)}{max(Lv - pos)} \tag{2}$$

- Lv-pos(w,si), as described in equation 1;
- max(Lv-pos),is the highest value of all L-scores in a document.

4.6 Summary Generation

The final step in our pipeline aims at generating an extractive summary by selecting the most important sentences from the original document. This step is formulated as a maximum coverage problem, where the goal is to choose the minimum subset of sentences that covers the most signicant concepts in the document. To achieve this, we rely on an ILP-based technique based on concepts. We employ the GNU Linear Programming kit [2] to solve the optimization problem. Our approach is inspired by the work of Gambhir and Gupta [11], who identified relevance, redundancy, and length as key features of a good summary: In other words, our aim is to select a subset of sentences that encompass the most relevant information from the document while avoiding redundancy and maintaining the summary within a desired length range.

[2] `https://www.gnu.org/software/glpk/`

$$max(\sum_{ci \in C} w_i c_i + \sum_{sj \in S} co_j s_j) \tag{3}$$

$$\sum_{sj \in S} l_i s_j \leq L \tag{4}$$

$$s_j Occ_{ij} \leq c_i \forall i, j \tag{5}$$

$$\sum_{sj \in S} s_j Occ_{ij} \leq c_i \forall i, j \tag{6}$$

$$c_i, s_j, Occ_{ij} \in 0, 1 \forall i, j \tag{7}$$

In Equation (1), c_i and s_j are binary variables representing a concept and a sentence, respectively. The binary variable Occ_{ij} indicates the presence of a concept in a sentence. The weight of each c_i concept in the set of C concepts is represented by the variable w_i. The co_j variable represents the cohesion of the sentences generated by the entity graph. The objective is to maximize the informativeness of the summary by maximizing Equation (1), while ensuring local cohesion between the sentences by satisfying Equation (2).

5 Experiments and Results

This section presents the experimental evaluation conducted for the proposed mono-document ATS task. It includes details about the corpus used, the evaluation metrics employed, and a comparative analysis of our system against others.

5.1 Datasets

The experiments were carried out on the widely used datasets from the DUC 2001 and 2002 competitions for mono-document summarization. Table 3 provides details about the selected datasets. In these datasetts, the provided golden summaries are abstractive,i.e., they were manually created by humans. Each document includes two golden summaries, each containing around 115 words.

Table 3: Dataset Distribution

Dataset	Golden Summaries	Documents	Sentences	Words
DUC 2001	Abstractive (Human)	309	11026	269990
DUC 2002	Abstractive (Human)	576	14370	384012

5.2 Evaluation Metrics

To assess the effectiveness of the our summarization system, we employed the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [15], a widely-used automatic evaluation metric in ATS. Specifically, ROUGE-N measures the n-gram recall between a candidate summary and a set of reference summaries. Mathematically, ROUGE-N can be expressed as:

$$Rouge-N = \frac{\sum_{S=0}^{rf} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S=0}^{rf} \sum_{gram_n \in S} Count(gram_n)} \tag{8}$$

1. n, number of grams;
2. rf, number of references summarys;
3. $Count_{match}(gram_n)$,the maximum number of n-grams co-occurring between a candidate summary and a set of reference summaries;
4. $Count(gram_n)$, the maximum number of n-grams occurring in the reference summaries.

Although ROUGE is widely used ATS, it has certain limitations. While it effectively evaluates the informativeness of the generated summary, it lacks in assessing the coherence and semantic analysis of the summary. Consequently, in conjunction with ROUGE, the inclusion of other metrics may be essential to achieve a more comprehensive evaluation of summarization systems. To ensure a fair comparison, the same parameters in ROUGE settings were employed for all the systems compared in the subsequent sections.

5.3 Evaluation of the proposed system

This section presents the results of our ROUGE evaluation for the summaries generated by the proposed system. We compared four concept scoring metrics discussed earlier and explored hyperparameters associated with utilization of the similarity matrix and the selection of the best concepts based on the AMR hierarchy level of the sentence. Table 4 show the results using the complete pipeline for the concept score metrics and the similarity matrix.

The best results for both the DUC 2001 and 2002 datasets were achieved using S-pos scoring metric, with scores of 46.21 (DUC 2001) and 49.3 (DUC 2002). This metric takes into account the level of the concept in the AMR hierarchy of the sentences in each document and assigns higher scores to concepts generated by the AMR graph, denoting their importance. To ensure a fair comparison, the same parameters in ROUGE settings were employed for all systems compared in this study. It is worth mentionning that, despite the fact that ROUGE is widely employed to evaluate ATS systems, it does have limitations when it comes to assessing the semantic coherence of the summaries.

This section focuses on exploring the hyper/parameters to optimize the data and achieve the best possible results. Table 3 provides an overview of the DUC datasets. In addition, Table 5 shows a statistical analysis based on the AMR

Table 4: Best Results select all levels and using Similarity Matrix- DUC 2001 & DUC 2002

	DUC 01		DUC 02	
Sumarizer	R1	R2	R1	R2
Lv-pos	45.37	16.05	48.84	19.66
S-pos	46.21	16.80	49.3	20.22
Word Frequency	46.1	19.94	48.7	19.7
TF-ISF	44.22	15.07	46.91	18.30

graphs generated for each sentence in the datasets. Specifically, Table 3 summarizes the general characteristics of the datasets, whereas Table 5 examines the AMR graphs and their associated statistics.

Dataset	Maximum Level	Lowest Average	Highest Average	Average
DUC 2001	24	6.31	13.9	9.026
DUC 2002	25	5.5	15.5	9.22

Table 5: AMR levels in the selected datasets

The Maximum Level field indicates the highest level (maximum height of the graph) reached by a sentence in the entire corpus (24 for DUC 2001 and 25 for DUC 2002). To gain a better understanding of the use of AMR, we calculated the average number of AMR levels per sentence for each document. "Lowest Average" represents the smallest average in the dataset, while "Highest Average" denotes the largest one. On average, the size of the AMR graph of the sentences in both datasets is approximately 9 levels (9.026 for DUC 2001 and 9.22 for DUC 2002).

One interesting hyperparameter is related to the level of AMR used for selecting concepts in a sentence. As this project adopted the "Core Semantic First" AMR, which prioritizes the main concepts closer to the root, we can limit the concept selection to a specific level in the AMR hierarchy. To compare results, we experimented selecting concepts up to level 8 (below the average sentence levels), up to -level 9 (average sentence levels), up to level 10 (above the average sentence levels), and selecting all concepts at all levels. Table 5 provides a detailed analysis of the AMR graphs generated by each sentence in the datasets.

Table 6 shows the results of this experiment, whithout employed the similarity matrix. On the DUC 2001 dataset, the best result was obtained by employing the Word Frequency concept scorer and selecting concepts up to level 8 in the AMR hierarchy (R1 46.38 and R2 17.11). For DUC 2002 dataset, the best result was attained with the S-pos concept scorer and selecting concepts up to level 9 in the AMR hierarchy (R1 49.22 and R2 20.20).

Establishing relationships between sentences is a crucial step in the proposed system, as described in Section 3.3. The similarity matrix plays a vital role in this

Table 6: Results up to a certain level and all levels of the AMR graph

		DUC 01		DUC 02	
Concept Score	AMR Level	R1	R2	R1	R2
Word Frequency	8	46.38	17.11	48.91	19.85
Word Frequency	9	46.27	19.96	48.73	18.17
Word Frequency	10	46.20	17.03	48.84	17.03
Word Frequency	all	46.06	16.83	49.24	16.83
S-Pos	8	46.00	16.91	49.20	20.08
S-Pos	9	46.03	16.85	49.22	20.20
S-Pos	10	46.12	16.89	49.15	20.05
S-Pos	all	45.88	16.53	49.26	19.96
TF-ISF	8	44.23	15.24	47.03	18.17
TF-ISF	9	44.17	15.25	46.88	17.98
TF-ISF	10	45.10	14.33	46.68	17.75
TF-ISF	all	44.49	15.32	47.07	18.11
Lv-pos	8	45.67	16.43	48.98	19.82
Lv-pos	9	45.07	16.20	48.93	19.71
Lv-pos	10	45.21	16.45	49.01	19.74
Lv-pos	all	45.75	16.25	49.12	19.88

process by identifying the most similar concepts between two distinct sentences, thereby enhancing their relationships. Experiments conducted without using the similarity matrix are presented in Table 6, whereas Table 7 displays the best results when employing the similarity matrix. However, employing this matrix introduces a trade-off between performance improvement and system running time, as the latter is negatively impacted.

Dataset	Concept Score	AMR Level	R1	R2
DUC 2001	Word Frequency	8	46.41	17.05
DUC 2001	S-Pos	8	46.31	16.92
DUC 2002	Word Frequency	8	49.13	19.17
DUC 2002	S-POS	all	49.30	20.22

Table 7: Best Results of the select levels using the Similarity Matrix (DUC 2001/2002)

5.4 Comparison with other ATS Systems

In this section, we compare the performance of the proposed system, using the parameter settings that yielded the best results in terms of ROUGE, with other ATS systems in the literature. The systems for comparison include Concept Based-ILP [17], Regression-Based-ILP [18], Classifier4j [15], HP-UFPE [9], System-T and System-28 which achieved the best results in DUC 2001 and 2002 competition, respectively. To ensure a fair comparison, all systems are evaluated in terms of the same ROUGE-2 metrics [13]. The standard ROUGE-2 approach was employed, with the removal of stopwords and stemming. The results were obtained for both unigrams and bigrams.

The proposed system outperforms the others in the DUC 2001 dataset by selecting concepts up to level 8 of the AMR Graph,employing both the similarity matrix, Word Frequency for concept scoring. The results of this comparison are summarized in Table 8. In terms of ROUGE-1 (unigram), our system achieved the top score of 46.41 and ranked fourth in ROUGE-2 (bigram), while Regression-Based-ILP [18] secured the best score of 21.10 in ROUGE-2.

Our system achieved the second-best result in R1 (unigram) with a score of 49.30, being outperformed only by Regression-Based-ILP [18] with a score of 49.78. Regarding the DUC 2002 dataset, our system shows superior performance by selecting all AMR concepts. This setting was based on the similarity matrix, the S-Pos scoring method introduced in this work. In terms of ROUGE-1 (unigram), our system achieved the second-best score of 49.30, narrowly surpassed by Regression-Based-ILP [18] with a score of 49.78.

Table 8: Comparison Results

	DUC 01		DUC 02	
System	R1	R2	R1	R2
Concept Based-ILP	45.00	20.05	48.90	23.42
Regression Based-ILP	46.37	21.10	49.78	23.92
Classifier4j	45.18	20.62	47.98	23.64
HP-UFPE FS	37.07	15.34	47.2	19.86
System 28	-	-	48.71	23.65
System T	43.21	18.80	-	-
Towards Coherent-ILP	45.00	16.30	47.36	20.96
Proposed System	46.41	17.05	49.30	20.22

6 Conclusion and Future Work

This paper introduced a method for extractive text summarization based on an AMR semantic parser that identifies the most significant words within sentences. The proposed system leverages the DAG structure generated by the AMR parser, which positions the most important words closer to the root of the graph. Building on the initial word selection process, we assigned scores to words based on their level/position in the AMR graph. Next, we introduced a sentence similarity matrix for each document that assigns higher scores to words exhibiting similarity, based on frequency-based metrics. To ensure the selection of summaries that are both informative and more cohesive, we employed the ILP technique.

As future work, we intend to design new word-scoring metrics based on the AMR graph, taking into account the contextual information in which a word appears. By considering the surrounding context, we can better capture the importance of each word to the overall meaning of the text. In addition, we plan to evaluate the effectiveness of our proposed system on other datasets in distinct domains. By testing our approach on different datasets, we can gain insights into the system generalizability and understand how it performs in various domains and text genres. These efforts will contribute to a deeper understanding of the strengths and limitations of our proposed method and its potential for broader applicability in the field of ATS.

References

1. Allahyari Mehdi, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. Text summarization techniques: A brief survey, 2017.
2. Ramos Juan et al. Using tf-idf to determine word relevance in document queries, 2003. Oguzhan Tas and Farzad Kiyani. A survey on automatic text summarization, 2017.
3. Gupta Vishal and Gurpreet Lehal. A survey of text summarization extractive techniques, 08 2010.
4. Goldstein Jade, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: Sentence selection and evaluation metrics, 1999.
5. Banarescu Laura. Abstract meaning representation (AMR) 1.2. 5 specification, 23, 2018. Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Herm- jakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation (AMR) 1.0 specification, 2012.
6. Banarescu Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking, 2013.
7. Cai Deng and Wai Lam. Core semantic first: A top-down approach for amr parsing, 2019.
8. Shibhansh Dohare, Harish Karnick, and Vivek Gupta. Text summarization using Abstract Meaning Representation, 2017.
9. Ferreira Rafael, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima, Steven J. Simske, and Luciano Favaro. Assessing sentence scoring techniques for extractive text summarization, 2013. URL http://www.sciencedirect.com/science/article/pii/S0957417413002601.
10. Flanigan Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. A discriminative graph-based parser for the Abstract Meaning Representation, June 2014. URL https://www.aclweb.org/anthology/P14-1134.
11. Gambhir Mahak and Vishal Gupta. Recent automatic text summarization techniques: a survey, 2017.
12. Hermann Karl Moritz, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend, 2015.
13. Lin Chin-Yew. ROUGE: A package for automatic evaluation of summaries, July 2004. URL https://www.aclweb.org/anthology/W04-1013.

14. Liu Fei, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. Toward abstractive summarization using semantic representations, May–June 2015. URL https://www.aclweb.org/anthology/N15-1114.
15. Lothian N. Classifier4j, 2003. URL http://classifier4j.sourceforge. net/.
16. Lyu Chunchuan and Ivan Titov. AMR parsing as graph prediction with latent alignment, July 2018. URL https://www.aclweb.org/anthology/P18-1037
17. Oliveira H., R. Lima, R. D. Lins, F. Freitas, M. Riss, and S. J. Simske. A concept-based integer linear programming approach for single-document summarization, 2016.
18. Oliveira H., R. Dueire Lins, R. Lima, F. Freitas, and S. J. Simske. A regression-based approach using integer linear programming for single-document summarization, 2017.
19. Flanigan J., Chris Dyer, Noah A. Smith, and Jaime G. Carbonell (2016) Generation from abstract meaning representation using tree transducers. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 731–739. The Association for Computational Linguistics, 2016.
20. Banarescu L., Bonial C., Cai S., Georgescu M., Gritt K., Hermjakob U., Knight K., Koehn P., Palmer M., and Schneider N. (2013) Abstract Meaning Representation for sembanking. In Proceedings of the 7th Linguistic Anno- tation Workshop and Interoperability with Discourse, pages 178–186. Association for Computational Linguistics, 2013.
21. Palmer M., Gildea D., Kingsbury P. (2005) The proposition bank: an annotated corpus of semantic roles", Computational linguistics, vol. 31, no. 1, pp. 71–106, 2005.
22. Konstas, I., Iyer, S., Yatskar, M., Choi, Y., and Zettlemoyer, L. (2017). Neural AMR: sequence-to-sequence models for parsing and generation. CoRR, abs/1704.08381.
23. Wang, C., Nianwen, X., and Sameer, P. (2015). A transition-based algorithm for amr parsing. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics.
24. Damonte Marco, Cohen Shay B and Satta Giorgio. An Incremental Parser for Astract Meaning Representation,2017.
25. Matthiessen, C. M. I. M., & Bateman, J. A. (1991). Text generation and systemic-functional & linguistics: experiences from English and Japanese. London and New York: Frances Pinter Publishers and St. & Martin's Press.
26. Lancia, Giuseppe & Serafini, Paolo. (2018). Integer Linear Programming.