An AMR-based Extractive Summarization Method for Cohesive Summaries

Pedro A. X. Silva[1], Rinaldo Lima[1], and Bernard Espinasse[2]

[1]Departamento de Computacao, Universidade Federal Rural de Pernambuco (UFRPE) Recife – PE – Brasil
[2]Aix-Marseille Université (AMU), LIS UMR CNRS, Marseilles - France.

***Abstract*—The main goal of automatic text summarization is condensing the original text into a shorter version, preserving the information content and general meaning. The extractive summarization, one of the main approaches for automatic text summarization, consists to select the most relevant sentences of a document, and generate a summary. This paper proposes a new mono-document extractive summarization method using a semantic representation of the sentence of a document expressed in AMR (Abstract Meaning Representation). In this method, AMR semantic representation is used to capture the most important concepts of each sentence (in core semantic terms), and a concept-based Integer Linear Programming (ILP) approach to select the most informative sentences improving both relevance and text cohesion of the summary. Two datasets proposed by DUC (2001 and 2002) were used to evaluate the effectiveness of our method on extrative summarirazion and commparing it with other state-of-the-art summary systems.**

***Index Terms*—AMR, Summarization, Extrative Sumarization, Cohesive Summaries**

I. Introduction

The need for more succinct information based on the central concepts requires research related to the field of automatic text summarization (ATS) [1]. According to Tas and Kiyani [2], main goal of automatic text summarization is "condensing the original text into a shorter version, preserving the information content and general meaning". Automatic text summarization can be classified into two types: extractive or abstractive. Abstractive summarization tries to understand the main concepts of a text and to express these concepts in human natural language, using natural language processing (NLP) techniques to interpret texts and find the new concepts and expressions that best describe it, generating a new shorter text that conveys the most important information of the original text document [3]. The main idea of the extractive summarization method is to list and select the most important sentences in a document. Such approach weigh statistically the words in the documen and, by its turn, assignning weights to the sentences. Finally, the selected sentences are concatenated according to their positions in the original document.

ATS is also classified with the quantity of documents that is analysed simultaneously, it can be mono-document or multi-document. The first mentioned part of a single document producing a final summary and the multi-document approach has more than one document to process and generate a single summary. Given the complexity of implementing the abstractive approach, the most widely used approach is the extractive approach. The extractive approach will be the focus in this paper. One of the problems that arises in relation to automatic text summarization is in checking the quality of the summaries generated by the system. An excellent summary should contain the most relevant information and should exclude redundant information, in addition to being consistent and understandable [4].

One of the limitations of the automatic summary evaluation systems is that they are more informative than cohesive, such as the ROUGE metric. The importance of cohesion in the text ensures harmony and logical connection between sentences, which contributes to a better understanding by the reader. Therefore, it is important to use humans not only to evaluate the information contained in the abstract, but also to evaluate whether the summery is cohesive. We believe that one way to improve the relevance and cohesion of an extractive summary is to take more semantics into account into the summarisation process, by using a semantic representation of the sentences of the document to be summarized.

This paper proposes to use a semantic representation, the Abstract Meaning Representation (AMR), in order to improve extractive summarization in relevance and cohesion. Indeed, such a semantic representation allows us to characterise the semantics of sentences and thus to take them into account in the choice of candidate sentences for an extractive summary. More precisely this method first uses AMR semantic representation to extract the most important concepts of sentences of a document to be summarized, and then uses a concept-based Integer Linear Programming (ILP) method to select the best sentences of this document according these concepts improving the cohesion of the summary. Two datasets proposed by DUC, that of 2001 and 2002, were used to test the effectiveness of this summary. In addition, the proposed solution is compared with other state-of-the-art summary systems reported in the literature.

This paper is organised as follows. In Section 2, we present related works on automatic summarisation using semantic representations and also works improving cohesion in extractive summarization. In Section 3, we briefly present the AMR semantic representation and its parsing. Section 4 presents in detail the proposed method of extractive summarization based on AMR representations of sentences of the document to summarize. In Section 5, we discuss the results obtained by this new method on reference corpora of documents in summarization (DUC 2001 and DUC 2002) and we compare them to the results of other extractive summarization methods on the same corpora.

II. Related Works

In this section we present related works on summarisation using semantic representations. This works mainly concern the abstractive approach, which is still difficult to implement and still remains a challenge. However, several works concern the extractive approach, the dominant approach, particularly in concept-based approaches improving the relevance of selected sentences and the cohesion of extractive summaries.

A. Toward Summarization Using Semantic Representations

The first work in the literature using semantic representation for abstractive summarization is [14]. This work uses Abstract Meaning Representation (AMR) and the proposed method consists of 3 steps to generate a summary classified as abstract: (1) Use an AMR parser to generate sentence graphs in a document (the JAMR parser (Flanigan et al., 2014)), (2) combine and transform all AMR graphs generated in step (1) into a single AMR graph, and (2) generate a text from the summary graph. In the work of Dohare et al. [7], an abstract summarization system was proposed where a new pipeline with an intermediate step is explored using Abstract Meaning Representation (AMR), the AMR parser chosen was version 2 of JAMR (Flanigan et al., 2014). The pipeline proposed in this article first generates the AMR graph of an input document, through which it extracts an abstract graph and generates summary sentences from this abstract graph. This work uses the CNN-Dailymail (Hermann et al., 2015) dataset. These works related to the abstractive summarization approach are very interesting, but is extremely heavy to implement for documents and sentences of a certain size and still remains a challenge.

B. Extractive summarization using Integer Linear Programming

The extractive approach is still the most widely used in automatic summarisation. In order to improve it, various works are interested in the semantics of the sentences of the document to be summarised. These include [17] that proposes a concept-based integer linear Programming approach for single-document summarization. More precisely, they propose an unsupervised concept-based approach for summarization of single documents using Full Linear Programming (FLP). In addition, a new weighting method combining coverage and position of sentences is proposed to estimate the importance of a concept, as well as a weighted distribution strategy that prioritizes sentences at the beginning of the document, if they have relevant concepts. The experiments were done in 3 datasets, in the DUC 2001-2002 and in the CNN corpus. We can also mention the work of H. Oliveira et al [18] that propose a regression-based approach using Integer Linear Programming for single-document summarization. A new regression-based approach using Whole Linear Programming

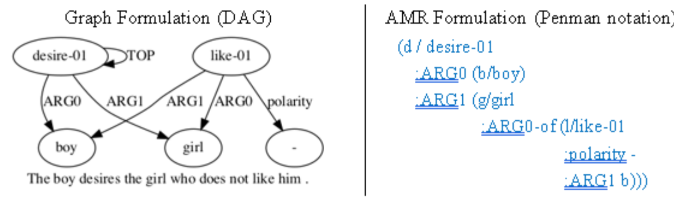

Fig. 1. AMR formulations of the sentence "The boy desires the girl who does not like him".

(ILP). Basically, the concept- based ILP method is used to generate several candidates abstracts for each input document, then a regression model is applied with several resources extracted at the summary, sentence and ngram level so that the most informative abstract is selected from the candidates. The experiments were carried out in the DUC 2001-2002 and CNN corpus. These works related to the extractive approach are interesting but do not seem to us to take sufficiently into account the semantics of the sentences of the document to be summarized, which would require the use of a true semantic representation of these sentences.

III. Abstract Meanning Representation (AMR)

The purpose of AMR is to capture the semantic meaning of the text, basically extracting "who is doing what with whom" in a sentence. AMR encodes, among others, information about semantic relationships, named entities, co-reference, denial and modality. The semantic representations can be considered as acyclic directed graph (DAG) labelled with root. AMR has now become a widely used formalism for the semantic representation of sentences, mainly for English. The main interest of AMR is to characterize the semantics of a sentence in an abstract representation by dropping several syntactic phenomena such as articles, number, time and voice of the verbs. For English, many tools are available to parsing a sentence in an AMR or generating a sentence from an AMR, using resources as PropBank. For other languages, these parser and generators are still under development. AMR [5] [6] captures the predicate-argument structure of a sentence, using external semantic resources such as lexicons. AMR is an effective representation for researchers work- ing on NLP tasks that involve the handling of semantic information for the automatic understanding and natural language generation. In AMR, verb semantics is based on the PropBank's annotation scheme [21].

A. AMR formulations

If the semantic representation of a sentence is based on the meaning of the words that compose it, a meaning that is generally explained in a lexicon, and AMR is placed at a higher level of abstraction by linking several meanings to the same concept, particularly for derived words. Thus, in English, the verb destroy and the noun destruction are represented by the same concept. The noun investor is represented using the same concept as the verb invest, based on the fact that the investor is the person who invests. Generally speaking, a representation of a sentence expressed in the AMR notation represents the meaning of a sentence by matching different grammatical realizations in a single representation. This representation can be formulated in 3 different formats: first-order logic (FOL), Directed Acyclic Graph (DAG) oriented with a single root, and textual format based on Penman's notation [22].

Fig. 1 presents AMR representations (DAG and Penman formulations) of the sentence " The boy desires the girl who does not like him".

B. AMR parsing

AMR is at the sentence level, this is because all sentences in an original document will be processed and a graph generated, it is valid to note that at the end of this step is generated AMR in the Penman formulation. Although AMR is a formidable semantic analysis tool, some main difficulties exist in AMR parsing.

First, the graphic structure is more complicated because of frequent re-entrances and non-projective arches. By example in Fig. 2, the sentence "The cat's desire is to eat fish" the root node is the concept "cat-01", one realizes that we have no way to access the node "desire-01" from the root. The re-entrances are given by the relations that have "-of", for example in Fig. 2, the edge "ARG0-of" creates a re-entrance.

Second, the nodes in the AMR have no explicit alignment of the original text tokens. As this paper concerns extractive summarization, it is necessary to use the original tokens of the sentence, AMR parser generates different nodes from the original words. For example, in Fig. 2 it is noted that the original sentence is "Eating fish is what the cat desires", the word "Eating" is mapped in the AMR graph as the node "eat-01".

There are 3 main classes of parsing models for generating from an English sentence its AMR representation: the graph-based models, the transition-based models, and the neural models.

- Graph-based models produce graphs that satisfy semantic fitness constraints according to a specific algorithm. These models first identify concepts from the sequence of words in a given sentence, and then identify the relationships between pairs of concepts by a Maximum Spanning Connected Subgraph (MSCG) algorithm. The best-known parser associated with these models is the JAMR system [10].
- Transition-based models aim at generating AMR graphs by converting syntactic trees of a given sentence. This conversion is done by applying specific transitions (actions) of maximum score on this tree. Unlike the previous model, this model tends to optimise AMR graphs according to local relations. The main parser associated with these models are the CAMR system [24] and AMR-Eager system [25] similar to CAMR but incremental.
- Neural network-based models are based on sequence-to-sequence or seq2seq models. A seq2seq model is based on an encoder and a decoder and is inspired by neural machine translation. The encoder takes the sentence elements as input and generates an intermediate vector representation. Then the decoder is trained both on the vector representation from the encoder and on the output sequence. In general the encoder and decoder are implemented by a neural network with memory units (RNN or LSTM). An analyser associated with these models is the Neural-AMR parser proposed by Kontas et al. [23].

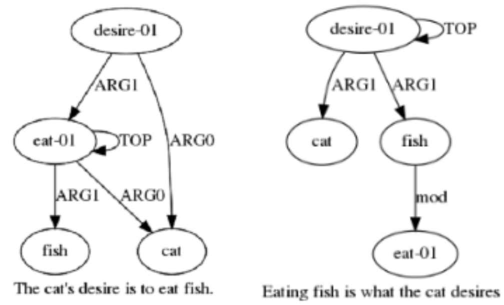

Fig. 2. AMR Graphs Examples

C. Capturing the semantic core of a sentence in AMR : Core Semantic First AMR Parser

In our extractive summarization context, capturing the "core semantic" of a sentence is very important. In AMR representation, the sentence graphs are organized in a hierarchy that the core semantics stay closely to the root of the graph, for which a top-down parsing scheme can fulfil the specific desiderata. However, according to Cai et al. [7], existing graph-based AMR parser cannot sufficiently model the interactions between individual decisions. The autoregressive nature of transition-based and seq2seq-based AMR parser leads to error propagation, where subsequent decisions can easily go awry, especially given the complexity of AMR. Consequently, Cai et al. propose [7] a novel AMR parser model known as Graph Spanning based Parsing (GSP), leading to top-down AMR parser permitting to capture the core semantic of the sentence. With this new AMR parser, called "Core Semantic First AMR parser", the most important words or concepts in the sentence are closer to the root of the generated graph, so the main semantic appears first in the graph. The Fig. 3 gives the DAG and Penman formulation obtained by this parser for the sentence "Congressmen to sue census over count of illegal aliens", and its core semantic are the root node "sue-02" (verb "to sue") and words or concepts nodes close of the root as census, person, congressman.

So, with the Core Semantic First AMR parser proposed by Cai and Lam [?] as the most important concepts of a given sentence are closer to the root of the AMR DAG of the sentence, it will be easy to extract the main concepts involved in each sentences of a document to summarize by traversing the AMR graph from the root to the bottom of the graph until a given level. Then we can compare different sentences according their extracted concepts, permitting a more relevant

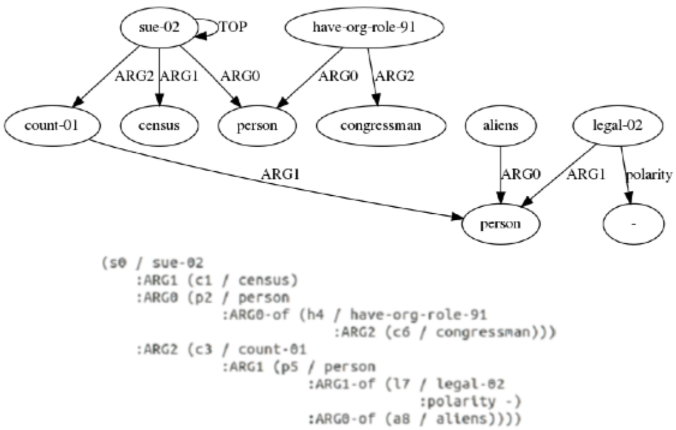

Fig. 3. AMR representation (DAG and Penman formulation) obtained by the "Core Semantic First" AMR parser proposed by [Cai and Lam, 2019] for the sentence "Congressmen to sue census over count of illegal aliens"

choice between candidate sentences for an extractive summary, as we explain in detail in the next section.

IV. An AMR-based Extractive Summarization Method for Cohesive Summaries

In this section we present the AMR-based Extractive Summarization method that we propose. First, we present its main process with its different step, then we present in detail each step of this process.

A. Main process

The AMR-based Extractive Summarization method that we propose is composed of several step as illustrated in the pipeline of the Fig. 4. The first step, AMR parsing step, is to parse in AMR the sentences of a given document to summarise. The second step, concept extraction step, consists to capture from the AMR graph of the sentence, the core semantic of each sentence, extracting a set of concepts from the AMR graph. The third step, sentences-concepts mapping step, define relations between sentences according the number of concepts in common, leading to define a sentences-concepts graph. In the fourth step, concept scoring step, each concept extracted is scored according to its importance in the text, permitting to list the sentences that have the best concepts. The last step, generation step, using scored concepts and relational sentence graph, concerns the generation of the extractive summary according an Integer Linear Programming (ILP) approach permitting to select the minimum subset of sentences to maximize coverage of important concepts.

B. AMR Parsing

This step use the Core Semantic First AMR Parser proposed by Cai et al. [7]. As presented in previous section, with this parser, the most important concepts of a given sentence are closer to the root of the AMR DAG of the sentence, it will be easy to extract the main concepts involved in each sentences of a document to summarize by traversing the AMR graph from the root to the bottom of the graph until a given level.

C. Concept Extraction

The main objective of this step is basically the reading and extraction of the concepts from the sentences represented in AMR format. To solve the problematic features presented in the previous topic, the recesses were disregarded and an algorithm was developed that reads the graph in the Penman formulation. To better understand this point of the approach, it is necessary to understand the architecture of the AMR graph. the parser generates a directed graph where the nodes are the concepts and the edges are the relationships between the concepts [20]. Some relationships are more important for the summary task, we developed a list of relationships called 'Stop-Edges'. There are 44 relations that were disregarded, for example the relations of date-entites (:day,:month,:year ...), of quantity (:quant,:unit,:scale) and some semantic relations (:consist-of,:age ...).

This step uses a rules-based algorithm to solve the problem of alignment of concepts in AMR and the original word in the sentence. However, there are some cases that need a specific treatment, a clear example is about polarity, notice that in the Fig. 5 the concept "appropriate" has a relationship with ":polarity -" this means that the original token is the negation of the concept "appropriate", ie the antonym: "inappropriate". The table I presents two sentences S1 and S2 and extracted concepts for each of these sentences with this rules-based algorithm.

TABLE I
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

D. Sentences-Concepts Mapping

The relationship between sentences is given by the number of concepts in common, generated by the sentences graph. In the sentences graph the sentences and concepts are nodes and the edges are directed from the sentence nodes to the concept nodes that exist in a sentence as in the Fig. 6. To increase the relationship between the sentences, a matrix was proposed with the similarity between all the words in the vocabulary of a document, so the words that are more similar are added in the sentences. The first step is to create a vocabulary with all the selected concepts. Table II present the set of extracted concepts from S1 and S2 constituent the vocabulary of the document consisting of sentences S1 and S2.

Then with this vocabulary we create the sentences-concepts graph. In the sentences-concepts graph 6.a of Fig. 6 where the green edges are the concepts that intercedes in sentences S1 and S2. After that, we select the concepts that are more similar. In the example proposed in the tables 1 and 2, it

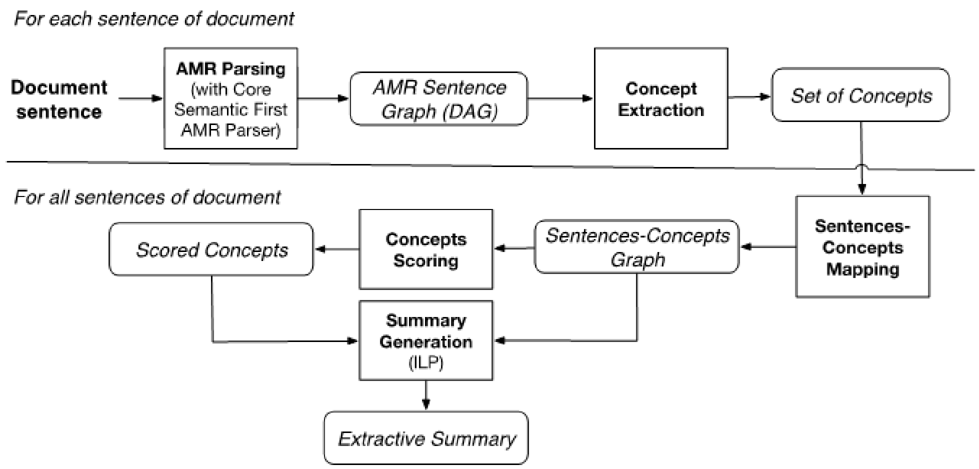

Fig. 4. Main process of the proposed method.

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

Fig. 5. AMR Penman Formulation of sentence: "the comment is inappropriate" (Banarescu, 2018)

TABLE II
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

is noticed that the words "Congressmen" in sentence 1 and "Legislators" in sentence 2 are similar to each other so a new sentences-concepts graph 6.b is generated (Fig. 6). Before using this approach, the relationship between sentences was only three concepts, now the relationship has been improved to five concepts.

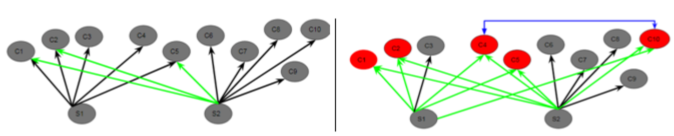

Fig. 6. Sentences graphs.

E. Concept Scoring

In our method it's necessary that each concept is scored according to its importance in the text, so we can list the sentences that have the best concepts. We used 4 concept score metrics, two of them with respect to the frequency of a concept in every document (Word Frequency and TF-ISF) and two with respect to the level of the concept in the AMR (Lv-Pos and S-Pos).

1) Position (Lv-Pos): Lv-pos is a metric that assesses the importance of a concept in the sentence based on the level of this concept in the AMR. Knowing that the Parser AMR that was used is in Top-Down architecture then the most important concepts are closer to the root [5]. The lower the level of this concept the more relevant this concept is to the system.

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

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

2) Sum of Level Position (S-pos): The S-pos is basically a normalized sum of all L-score concepts. Such a metric is at document level, analyzes the concept level in the AMR in all sentences, so the concept that appears most often and is closest to the root will be more relevant.

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

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

F. Summary Generation

The last step in this pipeline is the automatic generation of an extractive summary, i.e. the selection of the best sentences in an original document. This step is considered a maximum coverage problem, that is, selecting the minimum subset of sentences to maximize coverage of important concepts. For this, we use a concept-based Integer Linear Programming (ILP) approach, using the implementation of ILP available in GNU Linear Programming Kit to solve this optimization problem. According to Gambhir and Gupta [11] this approach tries to optimize 3 important features of a summary: (1) Relevance, (2) Redundancy and (3) Length. A subset of sentences covering the relevant text from the document collection is chosen.

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

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

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

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

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

In Equation (1), c_i and s_j are binary variables that are, respectively, a concept and a sentence and the binary variable Occ_{ij} indicates the presence of a concept in a sentence. While wi is related to the importance (weight) of each c_i concept in the set of C concepts. The co_j variable is given by the cohesion of the sentences generated by the entity graph. The main idea is to guarantee the informativity by the equation described in equation 3, while the equation 4 guarantees the local cohesion between the sentences.

V. Experiments and Results

This section aims to present the corpus used in the experimental evaluation, the metrics used for evaluation, and a comparative analysis of the results obtained by the tests performed by the proposed mono-document task of automatic text summarization.

A. Datasets

The experiments were done using the datasets widely used in the literature of the Document Understending Conference (DUC) 2001 and 2002 competition for mono-document summarization. Table III represents the information of the chosen datasets. We emphasize that the golden summaries are abstractive ones and were created by humans, in which a document has two golden summaries with approximately 115 words each.

Dataset	Golden Summaries	Documents	Sentences	Words
DUC 2001	Abstractive (Human)	309	11.026	269.990
DUC 2002	Abstractive (Human)	576	14.370	348.012

TABLE III
DATASET DISTRIBUTION

B. Evaluation Metrics

To evaluate the performance of the summaries generated by the proposed approach, we used an automatic evaluator most used in the literature, the Recall- Oriented Understudy for Gisting Evaluation (ROUGE). Formally, ROUGE-N is an n- gram recall between a candidate summary and a set of reference summaries. ROUGE-N is calculated 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.

Currently, the metric described above is among the main ones in the literature. However, there are some limitations when using it. The main goal is to evaluate the informativeness of the candidate summary, leaving it to be desired when it is a semantic analysis (cohesion) of the proposed summary in order to better evaluate the automatic text summarization systems. We highlight that, for all the systems compared in the following sections, we use the same parameters in ROUGE settings.

C. Evaluation of the proposed system

In this section, we will discuss the results of ROUGE for the summaries generated by the proposed approach. We will compare the four concept scoring metrics discussed earlier in IV (Word Frequency, Tf-ISF, Lv-Pos and S-pos), in addition to comparing other hyperparameters related to the use of the similarity matrix as well as the selection of the best concepts by level AMR of the sentence. Tables IV and V represent the results using the entire pipiline proposed for the concept score metrics and using the similarity matrix, these tables have 3 fields, (1) summary, (2) R1 (ROUGE -1, unigram) and (3) R2 (ROUGE-2, bigram). The best results for both the DUC 2001 and 2002 corpus were S-pos with **46.21** (DUC 2001) and **49.3** (DUC 2002). The reason is that this metric takes into account the level of the concept in the AMR of the sentences in every document, scoring the concepts generated by AMR graph as the most important.

Summarizer	R1	R2
Lv-Pos	45.37	16.05
S-Pos	**46.21**	16.80
Word Frequency	46.1	**16.94**
TF-ISF	44.2	15.07

TABLE IV
BEST RESULTS SELECT ALL LEVELS AND USING SIMILARITY MATRIX-DUC 2001

Projects of natural language processing (NLP) by the various stages and processes that this area needs, a large number

Summarizer	R1	R2
Lv-Pos	48.84	19.66
S-Pos	**49.3**	**20.22**
Word Frequency	48.7	19.7
TF-ISF	46.91	18.3

TABLE V
BEST RESULTS SELECT ALL LEVELS AND USING SIMILARITY MATRIX - DUC 2002

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 VI
AMR LEVELS DATASET STUDY

Concept Score	AMR Level	R1	R2
Word Frequency	8	46.38	17.11
Word Frequency	9	46.27	16.96
Word Frequency	10	46.20	17.03
Word Frequency	all	46.06	16.83
S-Pos	8	46.00	16.91
S-Pos	9	46.03	16.85
S-Pos	10	46.12	16.89
S-Pos	all	45.88	16.53
TF-ISF	8	44.23	15.24
TF-ISF	9	44.17	15.25
TF-ISF	10	45.10	14.33
TF-ISF	all	44.49	15.32
Lv-pos	8	45.67	16.43
Lv-pos	9	45.07	16.20
Lv-pos	10	45.21	16.45
Lv-pos	all	45.75	16.25

TABLE VII
RESULTS UNTIL CERTAIN LEVEL AND ALL LEVELS - DUC 2001

of hyperparameters and configurations are obtained. Thus, this section will explore these hyperparameters in order to better adjust the data to obtain the best results. Table III and Table VI represent a brief analysis of the DUC Datasets 2001 and 2002. More specifically, Table III deals with the general characteristics of the datasets used, while Table VI works with the statistics extracted from studies made using the AMR graphs generated by each sentence in the proposed datasets.

The Maximum Level field, is the maximum level (maximum height of the graph) of a sentence in the entire corpus (24 for DUC 2001 and 25 for DUC 2002), to better understand what occurs using AMR was done average AMR levels of the sentence per document. the Lowest Average field represents the lowest average of a document in the dataset and the Highest Average field the highest. Altogether, the overall average size of the AMR graph of the sentences for the two datasets is close to 9 levels (9,026 for DUC 2001 and 9.22 for DUC 2002).

An interesting hyperparameter is in relation to the AMR level in the sentence. As the AMR chosen for this project is the "Core Semantic First", the main concepts are closer to the root so we can select the concepts of a sentence up to a certain level in the AMR. From the study carried out in table VI, it can be seen that the number of levels in the datasets is on average approximately 9, so it was chosen, for comparison of results, up to level 8 (below the average levels in a sentence), up to level 9 (average sentence levels), up to level 10 (above the average levels in a sentence) and selecting all concepts at all levels. The results are shown in Tables VII and VIII, this experiment did not use the similarity matrix previously proposed. The best result of the DUC 2001 dataset was in R1 46.38 and R2 17.11, this system used the Word Frequency algorithm for the concept score and selected concepts up to level 8 in the AMR of a sentence. While for DUC 2002 the best result was in R1 49.22 and R2 20.20, it used the S-pos algorithm (proposed in this article) and selected the concepts up to level 9 in the AMR of a sentence.

The relationship between the sentences is a fundamental step in the proposed system, section 3.3 explains the similarity matrix whose main objective is to increase the relations between the sentences by the most similar concepts between two different sentences. Tables VII and VIII show experiments without using the similarity matrix, while Table IX shows the best results using the similarity matrix, it is worth noting that although there is an improvement when using the matrix the performance in relation to the system time is worse.

Concept Score	AMR Level	R1	R2
Word Frequency	8	48.91	19.85
Word Frequency	9	48.73	18.17
Word Frequency	10	48.84	17.03
Word Frequency	all	49.24	16.83
S-Pos	8	49.20	20.08
S-Pos	9	49.22	20.20
S-Pos	10	49.15	20.05
S-Pos	all	49.26	19.96
TF-ISF	8	47.032	18.17
TF-ISF	9	46.88	17.98
TF-ISF	10	46.68	17.75
TF-ISF	all	47.07	18.11
Lv-pos	8	48.98	19.82
Lv-pos	9	48.93	19.71
Lv-pos	10	49.01	19.74
Lv-pos	all	49.12	19.88

TABLE VIII
RESULTS UNTIL CERTAIN LEVEL AND ALL LEVELS - DUC 2002

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 IX
BEST RESULTS SELECT LEVELS AND USING SIMILARITY MATRIX - DUC 2001

D. Comparison with other Systems

In this section we will compare experiments of the proposed systems (The configuration that obtained the best result in ROUGE) with other systems available in the literature. The comparison systems are: Concept Based-ILP [17], Regression-Based-ILP [18], Classifier4j [15], HP-UFPE [9], System-T (Best result of the DUC 2001 competition), System-28 (Best result of the DUC 2002 competition). All systems used the same parameters as ROUGE-2 [13] as described in Table X.

Rouge type	Normal
Stop Word Removal	True
Stemmer	True
Ngram	1,2

TABLE X
ROUGE CONFIGURATION

In the DUC 2001 dataset the system proposed with the best result was with the configuration selecting the concepts up to level 8 of the AMR, using the similarity matrix and the Word Frequency algorithm for concept punctuation. The comparative results are in Table XI, The proposed system obtained the best result in R1 (unigram) with result in 46.41 and was fourth in R2 (bigram), the best result in R2 was Regression-Based-ILP 8371954 with 21.10.

System	R1	R2
Concept Based-ILP	45	20.05
Regression-Based-ILP	46.37	21.10
Classifier4j	45.18	20.62
HP-UFPE FS	37.07	15.34
System T	43.21	18.8
Towards Coherent-ILP	45	16.3
Proposed System	46.41	17.05

TABLE XI
COMPARISON RESULTS - DUC 2001

In the DUC 2002 dataset, the proposed system with the best result was with the configuration selecting all AMR concepts, using the similarity matrix and the S-Pos algorithm (proposed in this work) for concept punctuation. The comparative results are in Table XII, The proposed system obtained the second best result in R1 (unigram) with result in 49.30 second only to Regression-Based-ILP 8371954 with 49.78.

System	R1	R2
Concept Based-ILP	48.9	23.42
Regression-Based-ILP	**49.78**	**23.92**
Classifier4j	47.98	23.64
HP-UFPE FS	47.2	19.86
System 28	48.71	23.65
Towards Coherent-ILP	47.36	20.96
Proposed System	49.3	20.22

TABLE XII
COMPARISON RESULTS - DUC 2002

VI. CONCLUSION AND FUTURE WORKS

This paper presented a new method for extractive text summarization using the AMR semantic parser for identifying and classifyng the most important words in a sentence. The proposed system, in its best result, had AMR as a selector of the important features in the original text, because the construction of the acyclic directed graph (DAG) for the parser developed by (Cai and Lam, 2019) makes the most important words closer to the root of the graph. Aligned to this prior selection of the words, we were able to score the words by their level in the AMR graph. As a strategy to improve our results, we use a sentence similarity graph on every document to be summarized. Words that are similar may have higher scores, since we are relying on metrics based on frequency. Finally, for the choice of the sentences, we employed the ILP technique with the purpose of having more relevant and less redundant summaries. As future work we intend first to propose other word scoring metrics based on the AMR graph, as well as evaluate the effectiveness of the proposed system on other datasets enabling the direct comparision with other summarizers.

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 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. To- ward 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] 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.

[23] 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.

[24] 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.

[25] Damonte Marco, Cohen Shay B and Satta Giorgio. An Incremental Parser for Astract Meaning Representation,2017.