UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

A SURVEY ON SEMANTIC REPRESENTATIONS FOR TEXT SUMMARIZATION

MARCIO LIMA INACIO
THIAGO ALEXANDRE SALGUEIRO PARDO

Nº 436

RELATÓRIOS TÉCNICOS

São Carlos – SP
Dez./2021

A Survey on Semantic Representations for Text Summarization

Marcio Lima Inácio and **Thiago Alexandre Salgueiro Pardo**
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo
Avenida Trabalhador São-carlense, 400
São Carlos - SP, Brazil
marciolimainacio@usp.br, taspardo@icmc.usp.br

Abstract

Automatic Text Summarization in an important field of Natural Language Processing, which aims at selecting and presenting the most important information in a text, or set of texts, to the user. Many researchers advocate that using deep semantic representations produce better summaries, as they can leverage subtile language phenomena and knowledge. This survey presents an overview on the usage of explicit semantic representations for Automatic Text Summarization, including how they are specified and which methods were built upon them. The results analyzed indicate that deep semantics certainly improves summary quality, not only with structural, but also with linguistic information. The limitations of current automatic evaluation metrics for abstractive summarization is also briefly discussed, as lexical comparison metrics do not handle well abstracts, as they cannot deal with synonyms and rephrasing. Authors also consistently indicate that the quality of the representations play a large role in summary quality. These results open paths for future perspectives of Automatic Text Summarization and also explicit semantic representations.

1 Introduction

The goal of Text Summarization is to gather the most important pieces of information in a text – or collection of texts – and present them to the user according to their needs (Mani, 2001, p. 1). Summaries may vary in different aspects, e.g. the amount of information selected, their content and how they are presented to the user. These features are bound to the application in question, as well as to the users' demands.

Regarding the presentation of contents, summaries may be mainly classified into two types: extracts and abstracts (Mani, 2001, p. 6). Extracts consist of fragments of text taken directly from the original documents as they are, while abstracts contain rephrases of the original content, i.e. there are textual constructions different from the ones in the original text. An example of each summary format can be seen in Figure 1.

It is important to notice that the information unit for extractive methods is not necessarily a whole sentence (in the example, only some parts of the second sentence are presented), however all content presented is still taken directly from the original text. Nonetheless, it is most common to use the whole sentence as an extraction unit (Mani, 2001, p. 47), as taking sentence sub-units (phrases or words) may lead to more fragmented extracts.

With respect to abstracts, they may contain world knowledge, which is not included in the original text ("The Week column"). They also rephrase the initial content to avoid some redundancy, for example, connecting "John M. Fabrizi" from the first sentence to "Mr. Fabrizi" in the second one.

As described by Mani (2001, p. 18), summarization approaches can be classified according to the nature of linguistic knowledge they use. Shallower approaches rely on information bound to the text surface (its form, which words are used and how they are arranged syntactically). The author also indicates that there may be some semantic information, for example a method may use lexical semantic knowledge – such as ontologies or word embeddings – but whole sentences or texts are not evaluated in a deep semantic way.

John M. Fabrizi, the mayor of Bridgeport, admitted on Tuesday that he had used cocaine and abused alcohol while in office. Mr. Fabrizi, who was appointed mayor in 2003 after the former mayor, Joseph P. Ganim, went to prison on corruption charges, said he had sought help for his drug problem about 18 months ago and that he had not used drugs since. About four months ago, he added, he stopped drinking alcohol.

(a) Original text

John M. Fabrizi, the mayor of Bridgeport, admitted on Tuesday that he had used cocaine and abused alcohol while in office. Mr. Fabrizi said he had sought help for his drug problem about 18 months ago and that he had not used drugs since.

(b) Extract

The Week column. Mayor John Fabrizi of Brigeport, Conn, publicly admits he used cocaine and abused alcohol while in office; says he stopped drinking alcohol and sought help for his drug problem about 18 months ago.

(c) Abstract

Figure 1: Example of an extract and an abstract
Source: Adapted from Huang et al. (2020)

Deeper summarization approaches consider a more complex understanding of texts, usually using semantic representations to interpret their meaning. These methods surpass lexical semantics towards sentence and textual semantics – how concepts in a text relate to each other to produce meaning – and even including discourse-level information (how sentences in a text are related and how the main ideas are connected to create textual coherence and cohesion).

Numerous researchers – such as Mani (2001), Genest and Lapalme (2011), Li (2015) and Huang et al. (2020) – advocate that, to achieve good quality abstractive summarization, deep semantic approaches are required. Different arguments support this statement: Dohare et al. (2018) explain that shallow methods fail at dealing with some aspects of language, such as negation and coreference, which could be resolved through a semantic point of view. Another shortcoming of shallow methods is discussed by Greenbacker (2011), who states that semantic knowledge is important for multimodal processing, as all different media of information are related in an abstract level.

Sherry and Bhatia (2015) argue that a semantic representation is also important for multilingual summarization, since it can be used as an interlingua, connecting all information among different languages through meaning. This same sort of semantic interconnection can be used to gather important information scattered throughout the document in different sentences or paragraphs (Leskovec et al., 2005; Khan et al., 2016).

Another common argument for the inclusion of explicit semantic analysis for summarization is presented through the comparison with how humans produce abstracts. Leskovec et al. (2005) explain that humans create abstracts by interpreting the text, fusing its concepts and also rewriting some passages. Dohare et al. (2018) also defends that paraphrasing and the combination of concepts from the document are important to produce human-like abstracts.

All these ideas lead Genest and Lapalme (2011), as well as Khan et al. (2015), to state that using deeper approaches produce summaries that are more rich on information, more focused on important concepts and also less redundant.

Since there are multiple reasons to adopt deep summarization approaches – and, likewise, many works that do so – this report focuses on presenting the use of explicit semantic knowledge in those scenarios, more specifically on which semantic representations are used and how they are applied into the summarization process. We hope that this survey helps other researchers on understanding the potential and current limitations of these representations and their methods, as well as on supporting new works in the field.

2 Semantics

Semantics is the field of linguistics, which is dedicated to study meaning in language: how it is produced and perceived through linguistic form. There is a consensus that semantic knwoledge is governed by some rules, as speakers are able to interpret and extract the meaning from sentences never seen before rather than understand them by simply memorising each new one encountered (Kroeger, 2019).

There are many approaches to linguistic semantic analysis, such as *formal semantics*, *lexical semantics*, *cognitive semantics*, among many others, each of which conceptualising the idea of "meaning" from a specific perspective and also taking into account different aspects of language (Abrahão, 2018). For example, in *formal semantics*, "meaning" is described as being the relation between linguistic form and the things in reality that it describes. On the other hand, in *cognitive semantics* the meaning is accounted as a mental representation of the reality, i.e. the meaning does not represent reality *per se*, but a perception of reality for both the sender and the receiver of the message (Pinto et al., 2016).

In Natural Language Processing (NLP), the semantic analysis field is commonly named as *Computational Semantics*, comprising works related mainly to *lexical semantics* (Fellbaum, 1998; Navigli, 2009), *formal semantics* (Bunt et al., 1999; Blackburn and Bos, 2005) and *statistical semantics* (Sahlgren, 2008; Jurafsky and Martin, 2019). There are two main ways to computationally represent semantic knowledge: as numerical vectors learned from the words' co-occurrence – called implicit representations, distributional semantic models or also embeddings – or as formal explicit symbolic representations – also named metalanguages – in which the meaning is encoded as a specific data structure following definite rules determined by the designers of the representation.

Distributional Semantic Models exploit the meaningful information obtained from words and their contexts in text through frequency counts or even neural networks. Although these methods have been achieving state-of-the-art results in multiple tasks, there is still some uncertainty about how much semantic knowledge is actually acquired by the models from pure linguistic form without access to extratextual information (Bender and Koller, 2020).

As in this work we focus in explicit semantic representations, these should be explored further. According to Specia and Rino (2002), there are in NLP multiple approaches to represent meaning, each one seeking a common objective: precise definition, avoiding ambiguity, and that represents knowledge regardless of linguistic form (the actual text and words used), operating in an abstract level of meaning.

The construction of a semantic representation commonly follows the same workflow, starting from a linguistic theory, which defines how concepts should be interpreted and how specific linguistic phenomena should be handled. Afterwards, this theory leads to the linguistic specification of a semantic representation which is later implemented in a computational design taking into account particularities of this kind of model. As stated by Abend and Rappoport (2017), this type of representation is combined with information extraction methods in order to incorporate knowledge from outside the text surface.

Explicit semantic representations are primarily based on events and their argument structures (what are the entities and their specific relations to the events), as well as relationships between events or even discourse information. There are, however, many different ways to address each one of these aspects, resulting in multiple distinct representation guidelines (Abend and Rappoport, 2017). Thus, there is not yet a single universal semantic model that can be applied for unrestrained text contexts. The different representation formalisms used for text summarization and how they are exploited are presented in the following sections.

3 Representation Techniques

Semantic knowledge in Text Summarization research is mostly represented as graphs so that nodes represent concepts and edges relations between them. However, there are also other kinds of representations which do not necessarily take the form of a graph. As a common point, these representations are usually centred in representing verbs and their arguments (also known as thematic relations or semantic roles) (Chierchia, 2003). One can also argue that, more generally, all semantic representations deal with three main points: objects (or concepts), their properties and how these concepts relate with each other.

In order to sistematically organise these kinds of representations, Graph-based Representations are presented in subsection 3.1, while other forms of representations are introduced separately in subsection 3.2. Both sections are organised chronologically.

It is important to emphasise that these are not the only ways of explicitly represent semantic knowledge, as there are numerous semantic formalisms. Nonetheless, these are the representations used in research on Automatic Text Summarization, specially more recently.

3.1 Graph-based Representations

Back in the 90s the Universal Networking Language was created with the objective of serving as a language-independent semantic representation specially for Machine Translation systems (Uchida et al., 1999). Later, this representation has also been investigated for text summarization (Martins and Rino, 2001; Sornlertlamvanich et al., 2001; Mangairkarasi and Gunasundari, 2012).

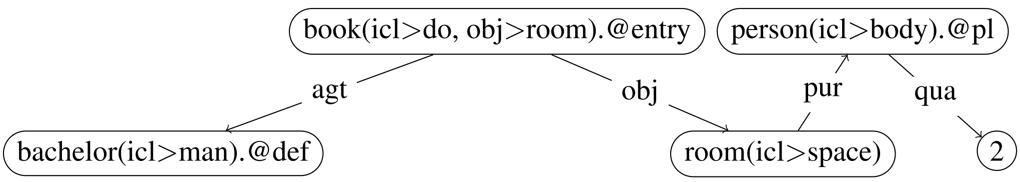
This representation is based on binary relations (called Relation Labels, RL) between concepts (called Universal Words, UW). Relation Labels are symbols representing semantic relationships such as *agt* (agent), *mod* (modifier), *obj* (object), *etc*. Universal Words are symbols (typically English words) which indicate generic concepts like *book* or *animal*.

Concepts in UNL can also include additional information, as in the UW room(icl>space), which indicates that the concept *room* is an hyponym of the concept *space*. UNL representation also includes Attribute Labels (AL), represented by an at sign (@), that represent attributes about relations and concepts.

As UNL representations are constituted by concepts and relations, they can be interpreted as graphs. An example for the sentence "The bachelor books a room for two people" can be seen in Figure 2.

obj(book(icl>do, obj>room).@entry, room(icl>space))
agt(book(icl>do, obj>room).@entry, bachelor(icl>man).@def)
pur(room(icl>space), person(icl>volitional thing).@pl)
qua(person(icl>volitional thing).@pl, 2)

(a) UNL relations



(b) UNL graph

Figure 2: Example of UNL relations and graph for the sentence "The bachelor books a room for two people"

Source: Adapted from Sornlertlamvanich et al. (2001)

The UNL representation is still sparsely used until this day, not only for summarization (Chaud and Di Felippo, 2018), but also for other purposes such as Machine Translation (Tomokiyo and Boitet, 2016).

Another early explicit semantic representation for multi-document text summarization was proposed by Vanderwende et al. (2004). They represented each sentence as a graph in Language-Neutral Syntax (LNS) format developed by Campbell and Suzuki (2002).

LNS represents a sentence as a tree, in which edges represent "deep grammatical functions" – which can be interpreted as semantic roles, as they indicate logical relations rather than purely syntactic associations – between concepts stored in nodes. An example can be seen in Figure 3, in which the sentence "Was the man persuaded to leave?" is represented as the corresponding tree.

Every content word from the original sentence is represented as a lemma in semantic head (*SemHeads*) nodes, while function words (such as articles) and other information (for instance, morphological inflection, verbal tense and voice, *etc.*) are represented as features (+*Past*, +*Pass*, +*Def* and +*Sing* in the

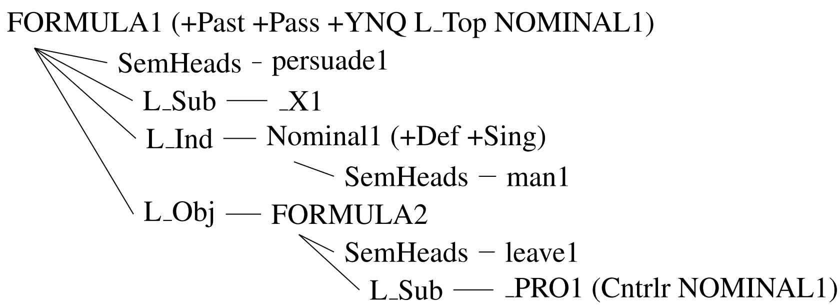


Figure 3: Example of LNS tree for the sentence "Was the man persuaded to leave?"
Source: Adapted from Campbell and Suzuki (2002)

example). Other semantic information, such as the type of the sentence – Yes/No question (*+YNQ*), wh-question (*+WhQ*), imperative (*+Imper*), declarative (*+Proposition*) – is also represented as a feature.

LNS trees also permit the definition of variables (*Cntrlr*) to link anaphoric pronouns with their antecedents. In the example, the man is both who has been asked and who leaves, so there must be an indication of this phenomenon. There are still multiple formulations for different linguistic phenomena (e.g. copula, modifiers, negation, *etc*.). For a complete description of the LNS representation, we suggest the reading of Campbell and Suzuki (2002).

As the LNS representation is sentential, Vanderwende et al. (2004) merge every graph in a group of documents in order to represent the meaning of the entire cluster. All identical nodes are joined together preserving their original relations with other nodes. If there are multiple occurrences of a relation between two nodes, the number of occurrences is stored in order to preserve its importance. An example of the final representation is given in Figure 4.

The LNS representation has also been used by Leskovec et al. (2004a; Leskovec et al. (2004b) and Leskovec et al. (2005), however the final graph representation is built in a different way than the one proposed by Vanderwende et al. (2004). This representation is obtained from Agent-Verb-Patient triples extracted for each sentence from the LNS representation. Then each triple is added to the final document graph by representing both the Agent and the Patient as nodes and the Verb as an edge connecting them.

They also apply coreference resolution in order to merge multiple mentions of the same entities under different names, e.g. "Tom Sawyer", "Tom" and "he", when referring to Tom Sawyer, would be represented as the same node. An example of this representation can be seen in Figure 5. The graph represents three sentences: "Tom Sawyer went to town", "He met a friend" and "Tom was happy".

It has also been reported in Leskovec et al. (2004b) the use of WordNet (Fellbaum, 1998) synsets to identify synonymy relations in order to join more relations and obtain a more compact representation, this process has been called semantic normalization. For example, both triples watcher-follow-moon and spectator-watch-moon should have the same representation as they have the same meaning.

This representation is also enriched with additional information in each node, mainly adjectives, which indicate more content about the concept being represented.

The same process has been used in Rusu et al. (2009) with the difference that Agent-Verb-Patient triples have been extracted from dependency syntactic trees rather than from LNS graphs. The authors applied also coreference resolution and semantic normalization, similarly to Leskovec et al. (2004b).

This is a representation that aims at being located in the edge between syntax and semantics, representing sentences from a semantic perspective (as it takes into account logical relations between the concepts), but is also highly biased towards syntactic form, since the authors want it to be possible to recover the full textual surface from a single LNS graph, which does need some syntactical and morphological information. Therefore, these graphs tend to look like dependency parsing trees, however they are still built in a logical level of relations instead of relying purely on the syntactic form.

Although LNS has already been explored for other tasks, e.g. Machine Translation (Brockett et al.,

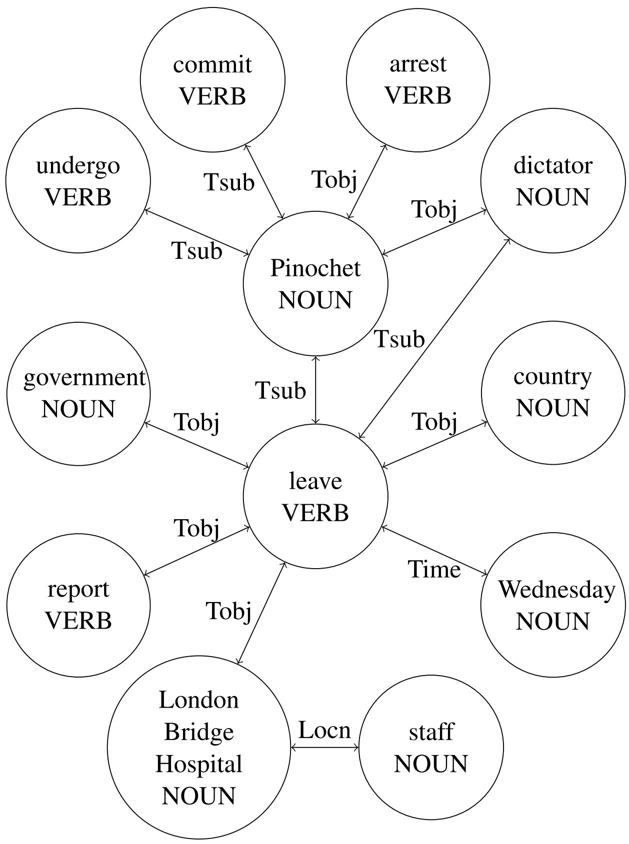


Figure 4: Example of merged LNS graph for a document cluster
Source: Adapted from Vanderwende et al. (2004)

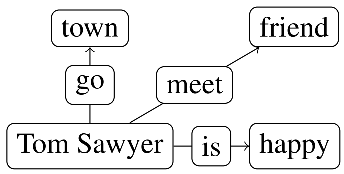


Figure 5: Example of a graph representation derived from LNS triples
Source: Adapted from Leskovec et al. (2004b)

2002), there has not been reported works using Language-Neutral Syntax during the last decade.

Another graph semantic representation used for automatic summarization has been described by Greenbacker (2011). This representation is produced using Sparser (McDonald, 1992), a chart-based constituent parser, which also contains an extensible semantic grammar. As usual, nodes represent concepts connected by (unlabeled) edges which indicate relations between them.

Each node also stores attributes, which indicate important information about the concept, and also references to the original excerpt of the text in which the concept occur. An example can be seen in Figure 6.

As Greenbacker (2011) focuses on multimodal summarization, this representation can also include semantic information extracted from images, audio or other types of media.

A limitation of this representation is that it relies on a manually-constructed domain ontology with all semantic relations to be extracted from the parsing trees of the document. There are, however, no further works that use this specific formalism.

A further ontology-based graph semantic representation, called Rich Semantic Graph (RSG), was

Figure 6: Example of nodes with attributes and references to original sentences
Source: Greenbacker (2011)

used by Moawad and Aref (2012). It contains nodes representing verbs and nouns along with edges corresponding to relationships between them. A domain ontology is used to create the sentence graph by instantiating nodes from concept categories, interconnect them and also validate the produced graph. This is a multi-sentence representation, however it does not represent necessarily a whole document, *i.e.* a document can produce multiple graphs from different sentences, but sentences which share information form a single graph, as some of their concepts are merged together.

An example of a RSG is presented in Figure 7. This graph represents the following sentences: “Angle Chris is a graduate student. Mrs. Chris is specialized in Machine learning field. Angle Chris published two papers in international conferences.”

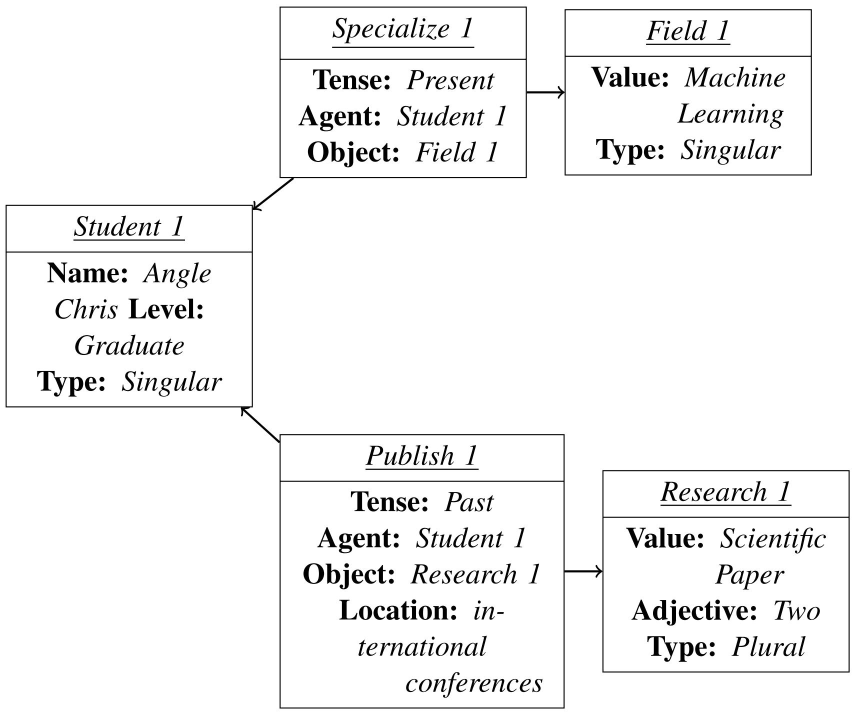
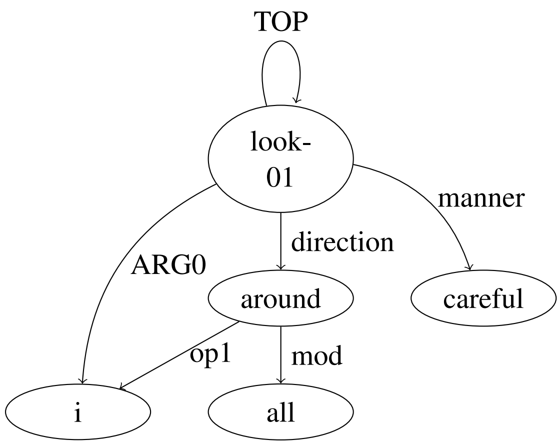


Figure 7: Example of a document in Rich Semantic Graph for the sentences “Angle Chris is a graduate student. Mrs. Chris is specialized in Machine learning field. Angle Chris published two papers in international conferences.”
Source: Adapted from Moawad and Aref (2012)

More recently, Abstract Meaning Representation (AMR) started being used in automatic text summarization (Liu et al., 2015; Dohare et al., 2018; Liao et al., 2018; Inácio and Pardo, 2021). This representation was created by Banarescu et al. (2013) based on well-consolidated resources, such as the PropBank (Kingsbury and Palmer, 2002).

In AMR, sentences are represented as directed acyclic graphs with nodes as concepts and labeled edges indicating relationships between them. These relationships can be inherited directly from the PropBank frames (indicating arguments of the verb) or special AMR relations, such as "modifier", "time" or "location".

Each AMR graph can be also represented using PENMAN notation, which is a parenthesized version of the graph. This notation, similarly to LNS previously mentioned, associates a variable with each note, enabling referencing the same entity across the graph. An example for the sentence "I looked carefully all around me" can be seen in Figure 8.



(a) AMR Graph

```
(l / look-01
   :ARG0 (i / i)
   :direction (a / around
      :op1 i
      :mod (a2 / all))
   :manner (c / careful))
```

(b) PENMAN notation

Figure 8: Example of AMR graph and PENMAN notation
Source: Adapted from Dohare et al. (2018)

AMR is a trending representation currently, with various works both parsing (Wang et al., 2016; Cai and Lam, 2020), text generation (Flanigan et al., 2016; Beck et al., 2018). There are also some applications of AMR besides text summarization, e.g. paraphrase detection (Issa et al., 2018) and event extraction (Rao et al., 2017). AMR is appealing because it may be considered simpler, since it does not commit itself to incorporate some information (such as verbal inflection or plural). Also, the annotation process has some degree of freedom, as there is not a specific restrictive set of steps to follow, only the guidelines to ensure that the basic structures of the representation are met. Alongside the usage of well-established resources, all these characteristics make the annotation process faster, according to Banarescu et al. (2013).

Meanwhile, another semantic graph representation for documents has been reported by Khan et al. (2016) and Khan et al. (2018). This representation is based on Predicate Argument Structures (PAS) (Khan et al., 2015). A PAS is a structure containing a verb and its arguments obtained automatically via Semantic Role Labeling (SRL). The authors considered only a limited set of roles: A0 (subject), A1 (object), A2 (indirect object), ArgM-LOC (location) and ArgM-TMP (time).

The graph is created by calculating the similarity of each pair of PASs according to Jiang's distance (Jiang and Conrath, 1997) between each corresponding element (e.g. A0 is compared only with the other A0). This distance uses the WordNet (Fellbaum, 1998) hierarchy to calculate the similarity between two concepts.

Afterwards, a fully connected graph is built so that vertices represent PASs and edges are weighted according to the computed similarity. Edges with a similarity below a given threshold are dropped, as well as loops (edges between a node and itself).

The edges weighting is also enriched with a similarity measure between the PAS and the document it pertains (Khan et al., 2016). This is done by considering weighted textual features such as the positioning of a PAS in the document or the overlap of the PAS with the title of the document. Optimal weights are obtained via Genetic Algorithm.

Subsequently, the weight of an edge between two nodes is updated taking into consideration the similarity of both PASs with their respective documents. The influence of each PAS in this update is controlled by a hyperparameter μ set beforehand.

In Khan et al. (2018), the authors also incorporated the similarity between a PAS and the set of documents to which it pertains. This process includes different features than the PAS-to-document similarity, such as frequent terms TD-IDF scores. Each feature also has an optimal weight obtained using Genetic Algorithm.

This kind of representation has an advantage of its creation being fully automatic, only limited by the quality of the SLR systems. As an aftereffect, this representation is more restricted and simpler when compared to other ones, representing just a limited amount of semantic information. PAS representations have not yet been further explored.

Another semantic representation using graphs has been reported in Li et al. (2016). This representation – called Event Semantic Link Network (ESLN) – is event-centered, as authors focus on the news domain, which centers lots of information on specific events.

ESLN graphs consist of nodes representing events (actions) or concepts. Every noun phrase is considered a concept, as well as named entities. Each node also contains attributes about the corresponding concept, obtained from a dependency parsing tree through relations such as appositional modifiers or adjectival modifiers.

To build an ESLN graph it is also important to apply coreference and pronominal resolution in order to enable a single node representation for a single concept. This way, the most representative reference of the concept is maintained as the node main concept and all attributes of other mentions are merged together into this individual node.

As ESLN representations are graphs, they also have edges representing semantic relations between events and concepts. Similarly to the PAS-based representation already discussed, the authors focused on a limited set of argument types: actor, receiver, time and location.

This representation also includes edges connecting events, indicating some relations such as: time, cause-effect, purpose, instrument, condition, sequence and attribution. The edge labels are obtained automatically via classification with features extracted from both events and concepts.

Figure contains an example of the ESLN representation for the sentence "Lawyer Morris Dees, who is representing Victoria Keenan after she was attacked by two guards in July 1998, introduced depositions to contradict the men's testimony."

This representation can be considered highly related to AMR, as both focus on events, however ESLN can be considered more simplified, as it does not convey the total amount of semantic information that can be encoded in an AMR graph, such as possession or quantities. This is also a representation that has not yet been further developed.

Recently, Huang et al. (2020) presented a shallower semantic representation based on Open Information Extraction (OpenIE) triples (Banko et al., 2008). OpenIE is a framework that focuses on automatically extracting from syntactic parsing trees entities from a text and relations between them.

From this set of relationship triples, Huang et al. (2020) built their own graph representation by first ignoring all triples with any of the arguments having more than 10 words. From those triples that share two arguments and the different ones overlap, only the longest one is retained. Then, all arguments are turned into graph nodes, connected by edges representing the relations. A final step merges all nodes referring to the same entities by using a coreference resolution tool.

This representation is somewhat related to the other graph ones based from tuples, such as PAS and ESLN. This OpenIE-based representation, however, is not enriched with additional layers of knowledge,

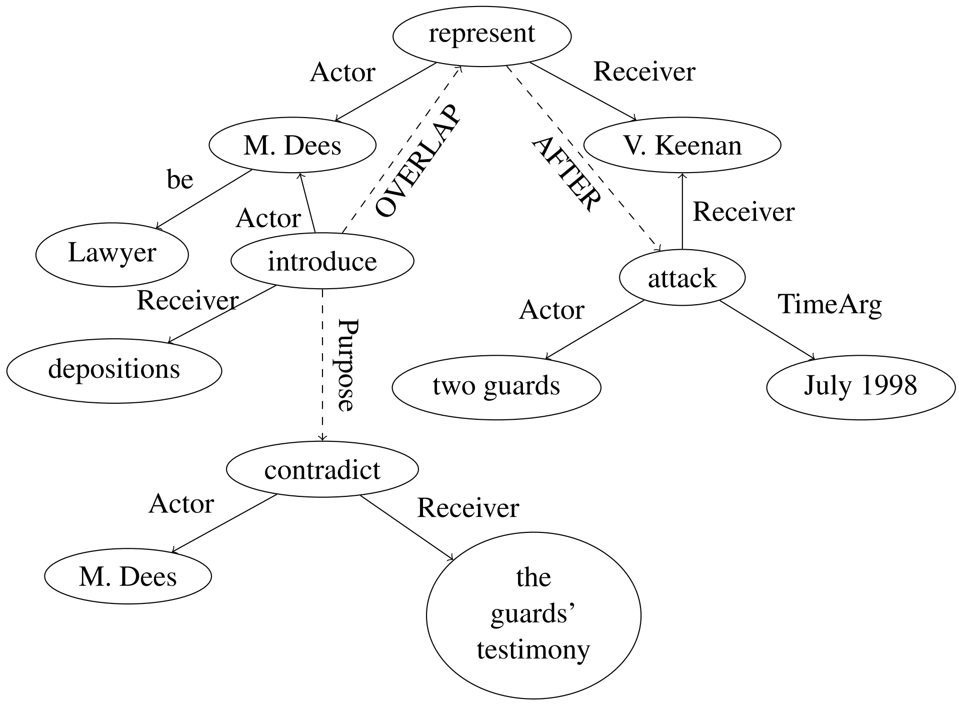


Figure 9: Example of ESLN graph for the sentence "Lawyer Morris Dees, who is representing Victoria Keenan after she was attacked by two guards in July 1998, introduced depositions to contradict the men's testimony."

Source: Adapted from Li et al. (2016)

such as time, location, cause-effect and others, which makes it to some extent shallower than the other ones.

3.2 Other Representations

Although graphs are an intuitive way of representing semantics of a text, there are works which represent this kind of knowledge in other formats. An example can be seen in Genest and Lapalme (2011), which introduce the concept of Information Items (InIts).

An InIt is defined as 'the smallest element of coherent information in a text or sentence' (Genest and Lapalme, 2011) and it can be as simple or complex as needed to convey the appropriate information. The authors acknowledge the vagueness of this definition and leave the implementation to be defined according to available resources, such as SRL, word-sense disambiguation, coreference resolution, *etc.*

As a simplified version in Genest and Lapalme (2011), InIts are implemented as syntactical subject-verb-object triples instead of abstract semantic information.

Later, a similar triple structure has been used by Li (2015) under the name of Basic Semantic Unit (BSU). A BSU consists essentially of an action together with an actor and a receiver. Later, this representation evolved into the ESLN graphs (described previously in subsection 3.1).

To build BSUs, the authors use Named Entity Recognition (NER) and coreference resolution. Afterwards, the triples are extracted from both constituent and dependency parsing trees by manually written rules.

It is also important to notice that the PAS representation, already explained previously, has also been used individually as a non-graph representation by Khan et al. (2015), also evolving later into a graph representation.

3.3 Overview

The main representation approach to semantic knowledge, among those presented in this work, is to use graphs; even those representations that are not explicit graphs may interpreted as being so, as they are mainly triples (*e.g.* InIt), or have evolved into later graph representations (*e.g.* BSU).

There are multiple intersections between representations, as they focus on conveying roughly the same type of information, even though it is in a more abstract level (UNL), or in a more text-grounded level (LNS). Some representations are practically the same, specially the triple ones (InIt, BSU and PAS), since they are generally simpler and based in actions, actors and recipients.

The representations by Greenbacker (2011) and Moawad and Aref (2012) (RSG) are also related, as both are oriented by a domain ontology created by a specialist, differing by the actual information derived from the ontologies (as each ontology is created for every specific domain, this varies also through applications of the same representation).

All representations presented are compiled in Table 1.

Table 1: Summary of all semantic representations studied

Representation	First publication	Type	Language specific?	External resources?
UNL	Uchida et al. (1999)	Graph	No	No
LNS	Campbell and Suzuki (2002)	Graph	No	No
Multimodal representation (no official name)	Greenbacker (2011)	Graph	No information	Yes, an ontology
InIt	Genest and Lapalme (2011)	Triples	No information	No
RSG	Moawad and Aref (2012)	Graph	No information	Yes, an ontology
AMR	Banarescu et al. (2013)	Graph	Yes	Yes, PropBrank
PAS	Khan et al. (2015)	Originally triples, evolving to graph	No information	No
BSU	Li (2015)	Triples	No information	No
ESLN	Li et al. (2016)	Graph (evolving from BSU)	No information	No
OpenIE-based (no official name)	Huang et al. (2020)	Graph (from triples)	No information	Yes, OpenIE and coreference resolution systems

4 Summarization Approaches

Many abstractive summarization methods rely on semantics, as they argue that this kind of summarization requires deeper textual comprehension in order to guarantee coherence in the output and also to deal with important linguistic phenomena, such as negations and rephrasing. There are multiple methods for summarizing texts using semantic knowledge. Some are deeply bound to the chosen representation whilst others are independent of the representation format. These methods are mainly based on three main approaches: score optimization, graph manipulation through rules and machine learning.

4.1 Score-based approaches

The most usual method to deal with explicit semantic representations for text summarization is to calculate a score of importance for the semantic units and then select the most important information to compose the summary. This has been explored for a large range of representations: UNL (Sornlertlamvanich et al., 2001; Mangairkarasi and Gunasundari, 2012; Sherry and Bhatia, 2015), LNS (Vanderwende et al., 2004), InIt (Genest and Lapalme, 2011), multimodal representations (Greenbacker, 2011), AMR (Liu et al., 2015; Dohare et al., 2018; Liao et al., 2018; Inácio and Pardo, 2021), BSU (Li, 2015), PAS (Khan et al., 2015; Khan et al., 2016; Khan et al., 2018) and ESLN (Li et al., 2016).

A rather simple scoring method is to determine the importance of some semantic concept by the number its occurrences within its corresponding document.

These counts have been used by Genest and Lapalme (2011) using the InIt representation of sentences. Although the authors say that the scoring should be done in a conceptual level (InIts), they obtained poor results using this approach. Thus, they calculated the score in the full sentence generated from the InIt.

The score for a sentence is computed as the average document frequency of each word lemma in the sentence, stop words and words already in the summary are ignored (this prevents redundancy). Finally, the sentences with highest score are selected to compose the summary one by one. Later, the sentences are ordered according to the time information stored in their InIt if it exists, otherwise the date of publication is used instead.

Mangairkarasi and Gunasundari (2012) also explored frequency counts to determine the importance of UNL concepts. A minimum threshold is set to define a concept as relevant to be included in the summary. Afterwards all sentences that contain the relevant nodes are selected as a summary.

The term frequency scoring can be further enhanced by incorporating the inverse document frequency to avoid words that are too frequent in many documents and may not be relevant to a single one (e.g. function words). This results in the TF-IDF (term frequency, inverse document frequency) score, as described in Equation 1.

$$\text{TF-IDF}(c_i) = \mathrm{c}(c, d) \cdot \log \frac{|D|}{\sum_{d' \in D} \delta(c, d')} \tag{1}$$

In the equation, $\mathrm{c}(c, d)$ represents the number of times concept c occurs in a given document d and D is a large corpus of documents from which d may or may not have been drawn. $\delta(c, d')$ indicates if concept c is within document d', returning 1 if true and 0 otherwise, hence $\sum_{d' \in D} \delta(c, d')$ represents the number of documents in D that contain c.

There are different variations to this score, for both the term frequency factor and the inverse document frequency one. A common variation is the inverse document frequency smoothing, used to avoid zero counts of document frequencies (when concept c is not in any document within D), as can be seen in Equation 2.

$$\text{TF-IDF}(c_i) = \mathrm{c}(c, d) \cdot \log \frac{|D| + 1}{1 + \sum_{d' \in D} \delta(c, d')} \tag{2}$$

This scoring method was used, for example, by Sornlertlamvanich et al. (2001) for summarizing documents through the UNL representation. The method consists of four steps: sentence scoring, sentence selection, redundancy elimination and sentence combination.

Firstly, a score is calculated for each sentence (in UNL format) in order to determine their importance for being included in the summary. This is computed as the sum of the TF-IDF scores for all Universal Words (concepts) within the sentence representation. Subsequently, the sentences with highest score are selected to proceed with the summarization.

The next step is to remove auxiliary words that are not essential for their respective head elements. These auxiliary concept can be obtained by UNL relations such as *mod*, *man* and *ben*. The words to be dropped are obtained according to a contribution score, as shown in Equation 3. This score takes into account both Universal Words (w_1 and w_2) linked through a relation l. If the result is below a predefined threshold, the word can be dropped, as this indicates that it does not contain enough relevant information to be maintained.

$$\text{contribution}(l(w_1, w_2)) = \frac{\text{TF-IDF}(w_1)}{\text{TF-IDF}(w_2)} \tag{3}$$

Afterwards, the obtained sentences have their corresponding UNL graphs merged in order to reduce redundancy and create a more fluent summary. This is done by joining nodes with same UWs into a single one. The authors also defined a threshold to merge sentences with no more than 15 words, to avoid long sentences, which can sound unnatural.

This work has been further developed by Sherry and Bhatia (2015) by adding a previous step to the score calculation, in which the authors remove irrelevant relations (such as time, location and instrument). The new method also removes words as "further", "because", "aditionally" as they argue that these are not necessary for the summary.

Another work that used TF-IDF scoring to determine the importance of concepts is the one developed by Dohare et al. (2018) upon the AMR representation. As a first step, the most important nodes are selected to be included in the summary according to their TF-IDF frequency, using the CNN-Dailymail corpus (Hermann et al., 2015), which contains 300 thousand newspaper articles, to calculate the document frequencies.

After the node selection, this method searches for the most important relations between them by selecting the shortest path between every pair of nodes. The paths must also be the closest to the root of the first sentence in which the two concept occurred together. This approach has been used because the authors state that the most important relation is present in the first sentence in which the concepts appear simultaneously. Every node and relation in each path is also included in the final summary.

This exact approach was later used by Inácio and Pardo (2021) for opinion summarization using AMR. The authors used, however, multiple features to calculate the scores for each node, including numerous graph centrality measures, which are going to be better detailed throughout this report, as more approaches are explained. The authors also propose using Simulated Annealing optimization (Kirkpatrick et al., 1983) to combine the different scores into a single importance measure for the concepts.

A more complex scoring method, PageRank (Brin and Page, 1998), has also been explored in the literature as well as many variants of it. The main idea of a PageRank algorithm is that the importance of a node is given by the number of connections it has, thus a node "votes" in favor of its neighbors, increasing their importance. The process is recursive, as the more importance a node has, the greater is its votes power.

This procedure can be formalized as in Equation 4, in which the PageRank score PR of a node n is calculated according to its neighbors L. The number of outgoing links $\text{C}(l_i)$ from each neighbor l_i is also taken into account, as well as a dampening factor d (Vanderwende et al., 2004).

$$\text{PR}(n) = (1 - d) + d \sum_{l_i \in L} \frac{\text{PR}(l_i)}{\text{C}(l_i)} \tag{4}$$

The PageRank algorithm is iterative, with scores initialized uniformly. The scoring process is executed several times until the score values stabilize, i.e. there is little difference in these values for each node.

A work that uses this algorithm for summarization of LNS graphs is the one by Vanderwende et al. (2004), in which it is used to determine the importance of each node to be included in the summary. As a following step, the author used the node scoring to assign the level of importance for each link in the semantic graph. This is done according to Equation 5.

$$\frac{\mathrm{N}(i \xrightarrow{rel} n)\,\mathrm{PR}(i)}{\sum_{l_i \in L} \mathrm{N}(l_i \xrightarrow{rel} n)\,\mathrm{PR}(l_i)} \tag{5}$$

In the equation, the weight for the edge $i \xrightarrow{rel} n$ is calculated according to the number of occurrences of the said relation $\mathrm{N}(i \xrightarrow{rel} n)$, which is already stored in the LNS graph, as previously described in subsection 3.1.

After scoring, the summarization process is executed by selecting the highest ranked nodes together with their highest ranked relations (along the nodes to which they link). The minimum importance level of a node is defined by a threshold.

Another work that is inspired by the PageRank algorithm is the one of Greenbacker (2011). First, an information density metric was proposed, which takes into account some factors, as the ratio and importance of the arguments for the concept. This allows the summarization to favor nodes that contain more complete information. The arguments for the concept is obtained from a specialized domain-specific ontology.

The metric also considers the importance of other concepts connected to the node, as well as document and rhetorical structure, such as information about the position of the information in the document and about phenomena that may highlight a specific concept (e.g. juxtaposing or its relation to the document title).

After the computation of scores for each node, a PageRank-based algorithm (Demir et al., 2010) is used in order to rank the concepts, selecting the best concept at a time. After each selection, the scores are recalculated in order to ignore redundant information.

Finally the original phrasings for each node are used in order to provide information for the summary realization. A path is chosen between important concepts, as a microplanning step, including other concepts if necessary. Afterwards, their parsing trees are merged together in order to produce the final text.

Lately, Khan et al. (2018) developed a novel PageRank-based method for text summarization named Improved Ranking Algorithm on Weighted Graph (IWGRA), defined as in the Equation 6.

In the equation, the score of a vertex (PAS) v_i is calculated recursively taking into account the IWGRA score of all vertices v_j that point to v_i ($\mathrm{In}(v_i)$) weighted by the edge weight between them (w_{ji}). The links between each v_j and all nodes $\mathrm{Out}(v_j)$ going out of it are also considered. A dampening factor d_p is also present in the model.

$$\mathrm{IWGRA}(v_i) = (1 - d_p) + d_p \sum_{v_j \in \mathrm{In}(v_i)} \frac{\mathrm{IWGRA}(v_j) \cdot w_{ji}}{\sum_{v_z \in \mathrm{Out}(v_j)} w_{zj}} \tag{6}$$

In terms of execution, this algorithm initializes all scores as 1 and updates them according to Equation 6 until convergence is attained.

After calculating the scores, a Maximal Marginal Relevance method is applied in order to select best PASs to be included in the summary. This approach focuses on selecting nodes with highest score while avoiding redundancy within the summary, *i.e.* avoiding PASs similar to ones that are already in the summary.

This work can be seen as an evolution of the authors' previous works (Khan et al., 2015; Khan et al., 2016). The PAS graph representation in this work is enhanced once more, as described in subsection 3.1. Furthermore, a new method to calculate the importance of nodes is presented, rather than their previous works based on clustering.

These previous works of Khan et al. (2015) and, later, of Khan et al. (2016) are based in a similarity score between PAS pairs organized as a matrix. As the PAS representation is a tuple, the similarity between two PASs is calculated as the combination of the similarity for each element (arguments, predicate, location and temporal arguments), as can be seen in Equation 7.

$$
\begin{aligned}
\mathrm{sim}(v_{ik}, v_{jl}) = &\mathrm{sim}_{\mathrm{pred}}(v_{ik}, v_{jl}) + \mathrm{sim}_{\mathrm{arg}}(v_{ik}, v_{jl}) + \mathrm{sim}_{\mathrm{tmp}}(v_{ik}, v_{jl}) \\
&+ \mathrm{sim}_{\mathrm{loc}}(v_{ik}, v_{jl})
\end{aligned} \tag{7}
$$

In the equation, the similarity between PAS k from sentence S_i (v_{ik}) and PAS l from sentence S_j (v_{jl}) is calculated. It is based on the similarity between the predicates ($\mathrm{sim}_{\mathrm{pred}}$), arguments ($\mathrm{sim}_{\mathrm{arg}}$), temporal ($\mathrm{sim}_{\mathrm{tmp}}$) and location ($\mathrm{sim}_{\mathrm{loc}}$) arguments each of which is calculated as indicated in Equation 8.

$$
\begin{aligned}
&\mathrm{sim}_{\mathrm{pred}}(v_{ik}, v_{jl}) = \mathrm{sim}(P_{ik}, P_{jl}) \\
&\mathrm{sim}_{\mathrm{arg}}(v_{ik}, v_{jl}) = \mathrm{sim}(\mathrm{A0}_{ik}, \mathrm{A0}_{jl}) + \mathrm{sim}(\mathrm{A1}_{ik}, \mathrm{A1}_{jl}) + \mathrm{sim}(\mathrm{A2}_{ik}, \mathrm{A2}_{jl}) \\
&\mathrm{sim}_{\mathrm{tmp}}(v_{ik}, v_{jl}) = \mathrm{sim}(\mathrm{Tmp}_{ik}, \mathrm{Tmp}_{jl}) \\
&\mathrm{sim}_{\mathrm{loc}}(v_{ik}, v_{jl}) = \mathrm{sim}(\mathrm{Loc}_{ik}, \mathrm{Loc}_{jl})
\end{aligned} \tag{8}
$$

As can be seen from the equations, the PASs are compared according to each element they contain: predicates (P), arguments (A0, A1 and A2), temporal and location arguments (Tmp and Loc, respectively). The similarity between two elements is calculated according to two different metrics: Jiang's similarity measure (Jiang and Conrath, 1997) and edit distance. The former is used to compute the similarity between predicates and arguments and the latter as the similarity between temporal and location arguments.

Jiang's similarity measure uses the WordNet (Fellbaum, 1998) hierarchy to determine how related two concepts (C_1 and C_2) are. This is done by finding the least common subsumer (**lcs**) between them. Then, the similarity is calculated according to Equation 9.

$$
\mathrm{Jiang}(C_1, C_2) = \mathrm{IC}(C_1) + \mathrm{IC}(C_2) - 2 \times \mathrm{IC}(\mathbf{lcs}(C_1, C_2)) \tag{9}
$$

The similarity is calculated using the information content of each concept, as well as, the least common subsumer. This information content measure is calculated as shown in Equation 10, in which $\mathrm{freq}(C)$ represents the number of occurrences of concept C and N the total number of concepts.

$$
\mathrm{IC}(C) = -\log \frac{\mathrm{freq}(C)}{N} \tag{10}
$$

After the creation of the similarity matrix, all values are normalized into the range $[0, 1]$ and the diagonal is set to 0. Then, an agglomerative hierarchical clustering algorithm is performed on this matrix by first considering each element as a single cluster and iteratively merging similar structures.

Afterwards, a single element is selected from each cluster as the most representative to be included in the summary. This is performed by calculating a score considering 10 (ten) features about the PASs, which include structural (*e.g.* position in document) and linguistic (*e.g.* number of nouns and verbs) information. The score for a PAS P_i is calculated, as can be seen in Equation 11, as the linear combination of the features $\mathbf{f}(P_i)$ with weights $\mathbf{w}$. These weights are optimized via Genetic Algorithm.

$$
\mathrm{score}(P_i) = \mathbf{w}^{\intercal}\mathbf{f}(P_i) \tag{11}
$$

As can be seen, the similarity matrix obtained during this process can be interpreted as an adjacency matrix for the PAS graph representation described in subsection 3.1. This interpretation, among other enhancements on the representation, resulted in a new version of this work, reported in Khan et al. (2016).

Liu et al. (2015) also used a scoring method based on features and weights optimization. They worked, however, over the AMR representation. First, they create a multidocument AMR by creating a dummy root node and edges between each sentence root to this new root. This guarantees that the graph is connected. Then, concepts are merged together and multiple edges between two nodes are collapsed into one single unlabeled edge. The authors also expanded the edges in order to fully connect all nodes within the same sentence.

The summarization process is formalized as an Integer Linear Programming (ILP) problem that scores each possible subgraph, with vertices V' and edges E', according to Equation 12. Each vertex v in represented as a feature vector $\mathbf{f}(v)$, similarly each edge e is represented as $\mathbf{g}(e)$. Both θ and ψ are parameters to be estimated in the model through supervised learning.

$$\text{score}(V', E'; \theta, \psi) = \sum_{v \in V'} \theta^\intercal \mathbf{f}(v) + \sum_{e \in E'} \psi^\intercal \mathbf{g}(e) \tag{12}$$

The features used to represent concepts take into account their frequencies, their position within the text, their depth in the AMR graph, among others. Regarding the edges, they also compute their frequencies and position, but also use the features of the nodes in both ends of the edge.

The objective of the ILP model is to obtain the set of nodes and edges from the original multidocument AMR graph which maximizes the scoring. Some constraints have been incorporated in the model to also guarantee that the summary graph is connected.

Although AMR graphs are not necessarily trees, *i.e.* they may contain reentracies, the authors added a constraint to assure that the graph obtained in a tree, forcing every node to have, at maximum, one parent. There is also a last optional constraint that controls the size of the summary being produced in terms of the number of edges selected.

This method trains the weight vectors (θ, ψ) using a structured perceptron approach, which takes into account the structure of the selected graph so that it is close to a gold summary graph.

Later, Liao et al. (2018) enhanced this work by first clustering all sentences according to their content before selecting the summarized subgraph. This sentence clustering is done according to a similarity score. Some scores have been compared in the work: (i) Longest Common Subsequence (LCS); (ii) Vector Space Model (VSM), which uses cosine similarity between sentence vectors; (iii) Smatch (Cai and Knight, 2013), obtained from AMR graphs; (iv) Concept Coverage, prioritizing AMR representations with the same concepts. After clustering, some representative sentences are selected from each group in order to continue with the summarization process using same method as Liu et al. (2015).

This same strategy of finding, through ILP, a subgraph that optimizes a scoring function was explored by Li et al. (2016). The authors in this work use the ESLN representation, as they focus on summarizing news texts, in which events are the most important information concepts. This work, however, uses different features for representing nodes and edges than the ones used by Liu et al. (2015) and Liao et al. (2018).

After selecting the most suitable summary subgraph, the authors generate sentences automatically for each event in the graph using the SimpleNLG tool (Gatt and Reiter, 2009). Then, they merge two events if they share any semantic link, in order to create one single sentence according to multiple patterns. Events with the same actor are also merged using additive conjunctions.

To avoid problems in the automatic language generation process, the authors create numerous sentences with the tool and select the best ones in a greedy fashion, also avoiding redundancy in the summary. To do so, the linguistic quality (LQ) of a sentence s with L words must be calculated according to Equation 13.

$$\text{LQ}(s) = \frac{1}{1 - \frac{\log_2 \prod_{t=1}^{L} P(w_t | w_{t-1} w_{t-2})}{L}} \tag{13}$$

This linguistic quality measure is computed according to a trigram language model $P(w_t|w_{t-1}w_{t-2})$, which can be trained in large natural corpora to ensure fluency in the sentence.

A further scoring method of semantic representations was developed by Li (2015). This work performs summarization using BSU representations based on two metrics: semantic relatedness between BSUs and a salience score. The semantic relatedness between two BSUs is calculated according to three factors:

- Argument Semantic Relatedness (ASR)

This metric computes the relatedness between two concepts according to Wikipedia-based[1] Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007). This method is used to map text concepts into Wikipedia articles, then each article is represented as a numerical vector of TF-IDF scores of the words in it. Then these vectors can be compared through some similarity metric (*e.g.* cosine). The authors argue that the usage of encyclopedic texts are more fit to represent concepts, as they are generally written with the intent of defining explicitly the subject.

- Action Verbs Semantic Relatedness (VSR)

  This metric is computed using WordNet, as formulated by Mihalcea et al. (2006). Although six different WordNet-based metrics are presented by Mihalcea et al. (2006), the authors do not specify which one they used specifically to calculate this relatedness score.

- Co-occurrences in the Same Sentence (CSS)

  This measures if two BSUs occurred together in the same sentence or not.

Finally, the relatedness between two BSUs is obtained by linearly combining all three semantic relatedness metrics, however the author does not explain further how this combination is done.

Another score that is taken into account during summarization is the summary-worthy score. This measures how likely a given BSU is to be incorporated in the final summary. This score is obtained via Support Vector Regression (SVR), which is trained to assign a real-valued score between 0 and 1 for the input, so that lower values indicate BSUs that are not suitable to be in the summary.

After all, both values are used to calculate the final salience score for a given BSU_i according to Equation 14. SW represents the summary-worthy score, while R represents the semantic relatedness between two given BSUs.

$$\mathrm{saliency}(\mathrm{BSU}_i) = \mathrm{SW}(\mathrm{BSU}_i) \times \sum_j \mathrm{R}(\mathrm{BSU}_i, \mathrm{BSU}_j) \tag{14}$$

In order to select the best BSUs for the summary, a clustering of all BSUs is performed so that each cluster contains semantically similar BSUs. Then, the BSU with highest salience score within a cluster is selected as the representation of its group.

4.2 Rule-based approaches

An UNL-based summarization model, named UNLSumm, was presented by Martins and Rino (2001) and later extended in Martins and Rino (2002a) and Martins and Rino (2002b) by the same authors. This system consists of a set of 58 heuristics that are applied in each UNL graph in order to prune (i.e. eliminate) unimportant information.

The authors divide their heuristics into two groups: single and chained pruning. Single pruning rules are applied into a single relation from the UNL graph, while chained pruning acts upon a group of relations.

As this is a rule-based approach, the order in which the heuristics are applied is important and may affect significantly the summarized outcome. To address this problem, the authors decided to use the precision of each rule to determine its priority. The precision of a rule is defined as the probability of an application of the said rule lead to a satisfactory result, which retains the main idea of the original text.

Later, this method has been firstly enhanced by Martins and Rino (2002a), resulting in a total of 84 heuristics. An example of pruning can be seen in Figure 10, in which the sentence "Juliana went to college by car" is pruned into "Juliana went to college" by a rule that removes the "met" (method) relation.

Later, the system has been once more revised in Martins and Rino (2002b) resulting in two major changes in the summarization process. The first change excludes all rules over *mod* and *aoj* relations. The second modification changes the priority system, as the authors showed that mixing single with chained pruning heuristics led to poorer results.

[1] https://wikipedia.org

agt(go.@past, Juliana)
plt(go.@past, college)
met(go.@past, car)

(a) Before pruning

agt(go.@past, Juliana)
plt(go.@past, college)

(b) After pruning

Figure 10: Example of UNL pruning using heuristics
Source: Adapted from Martins and Rino (2002b)

A more recent approach based on rules is the one described by Moawad and Aref (2012). This work summarizes a text by reducing the original text RSG through a set of heuristic rules. These heuristics are created based on WordNet (Fellbaum, 1998) relations, such as hypermym, entailment and holonym. Each rule can merge, delete or consolidate certain nodes in the graph.

Each rule operates upon a pair of, so called, "sentences" $S_i = (\mathrm{SN}_i, \mathrm{MV}_i, \mathrm{ON}_i)$, in which SN (subject noun), MV (main verb) and ON (object noun) embody nodes from the original RSG. A heuristic example can be seen in Figure 11. This rule indicates that the main verbs and object nouns must be merged if they are similar to each other and if the subject relating to them refer to the same noun.

If	SN_1 is instance of noun N	and
	SN_2 is instance of noun N	and
	MV_1 is similar to MV_2	and
	ON_1 is similar to ON_2	and
Then	merge both MV_1 and MV_2	and
	merge both ON_1 and ON_2	

Figure 11: Example of heuristic for RSG reduction
Source: Adapted from Moawad and Aref (2012)

From this reduced RSG representation, the authors developed a text generation method (Fathy et al., 2012). To generate a text, they first exploit the WordNet structure to generate a list of synonyms for each node in the graph, weighting these synonyms according to WordNet rankings and group similarities. Then a discourse structuring process is carried out by creating pseudo-sentences for each noun according to their attributes and which verbs are connected to them.

Following, all pseudo-sentences are combined to form complete sentences, relying heavily in the domain ontology to retrieve the discourse relations needed. This process merges relating subjects as well as predicates with sharing information. Finally, the system organizes all sentences into full paragraphs, adjusting verbal inflection and punctuation as needed.

This rule-based method has been later explored for the Hindi language by Subramaniam and Dalal (2015).

4.3 Machine Learning approaches

Leskovec et al. (2004a) developed the first work using explicit semantic representations that used Machine Leraning algorithms for text summarization. This was accomplished by classifying subject-predicate-object triples extracted from the semantic graph (in LNS) according to their relevance to be included in the summary.

The authors developed a set of 466 features including linguistic knowledge (*e.g.* part of speech tags, named entity recognition tags), graph properties (*e.g.* PageRank scores, size of connected components) and other features, such as the position of the sentence in the text or the number of senses that a word has.

Then, a SVM classifier is trained to select relevant triples. Each sentence containing at least one relevant triple is selected to be included in the final summary. The same model has been reported by Leskovec et al. (2004b; Leskovec et al. (2005) and Rusu et al. (2009).

Leskovec et al. (2005) also report a comparison of the importance of each type of features used, showing that including information about the semantic graph structure does improve recall. They state, however, that these features are not sufficient to result in a good performance, as including further positional and linguistic attributes seems to enhance the results even more.

The authors also show some discrepancy between the ROUGE score and relevant triples F1 Score, since using specifically a linguistic feature set performs better according to the former, but not to the latter.

More recently, Huang et al. (2020) explored the usage of encoder-decoder Neural Network architectures for abstractive text summarization. The authors used a double encoder techquine, in which two encoders – one for the text *per se* and another for the semantic OpenIE-based graph – provide a combined representation of the input. Then, this representation is passed to a decoder module that produces the final summary text tokens.

Comparing their results with other ones using neural networks and language models, Huang et al. (2020) show that human judges consider that their deep semantic approach does produce more informative and fluent summaries. Their method, however, produces some hallucination errors, in which the system produces some content that was not present in the original text.

Inácio and Pardo (2021) also investigated the usage of Machine Learning classification (through classical methods such as Random Forest and SVM) to select important nodes from a graph based on some graph centrality and node frequency features, with a following rule-based selection of edges (similar to Dohare et al. (2018)). The authors also analyzed how this same classification approach behaves in selecting relevant relations through Levi Graphs; the results were, however, outperformed by using the edge selection rule set.

As a main contribution, the authors show that using the node classification approach, with the Random Forest algorithm, produced the best results for opinion summarization, outperforming even score-based approaches.

4.4 Overview

There are various approaches to exploit semantic representations for text summarization. These are primarily divided into three categories: score-based, rule-based and machine learning. Most works (14 out of 24) are based in some kind of scoring of semantic units (triples, subgraphs, nodes, *etc.*), while lesser works are focused on the application of rules (4) to determine which pieces of information are irrelevant (and can be dropped or merged with other parts of the text) and machine learning (4) to detect those elements worth of being in the summary.

There are multiple scoring methods that can be applied, from simple frequency counts (TF or TF-IDF) to more robust graph scoring methods (PageRank and ILP). Scoring approaches using further resources have also been explored, such as Jiang's Similarity Measure – which uses the WordNet – and Explicit Semantic Analysis, which uses Wikipedia (or any other encyclopedia).

All summarization methods described are summarized in Table 2, organized chronologically.

5 Final considerations

This work presented an overview of the usage of explicit semantic representations for automatic text summarization, presenting the arguments that motivated researchers into this approach, as well as describing all analyzed representations alongside their similarities and differences. The methods, which exploited these representations and the information they provide, have also been surveyed and categorized.

With respect to the impact of incorporating this sort of semantic knowledge into the summarization process, there is a general agreement that it does enhance summary quality in terms of multiple aspects. For example, Huang et al. (2020) compared human evaluations for methods with and without explicit semantic representations and those which take into account the representations and their structure contained more important information. This has also been observed by Li (2015), who states that their

Table 2: Summary of all text summarization methods studied

Publication	Type	Method	Representation used	Derived from
Martins and Rino (2001)	Rule-based	Pruning through rules	UNL	
Sornlertlamvanich et al. (2001)	Scoring	TF-IDF	UNL	
Martins and Rino (2002b)	Rule-based	Pruning through rules	UNL	Martins and Rino (2001)
Martins and Rino (2002a)	Rule-based	Pruning through rules	UNL	Martins and Rino (2002b)
Leskovec et al. (2004a)	Machine Learning	SVM	LNS	
Leskovec et al. (2004b)	Machine Learning	SVM	LNS	Leskovec et al. (2004a)
Vanderwende et al. (2004)	Scoring	PageRank	LNS	
Leskovec et al. (2005)	Machine Learning	SVM	LNS	Leskovec et al. (2004a)
Rusu et al. (2009)	Machine Learning	SVM	LNS	Leskovec et al. (2004a)
Genest and Lapalme (2011)	Scoring	TF	InIt	
Greenbacker (2011)	Scoring	PageRank	Multimodal	
Mangairkarasi and Gunasundari (2012)	Scoring	TF	UNL	
Moawad and Aref (2012)	Rule-based	Pruning/merging through rules	RSG	
Khan et al. (2015)	Scoring	PAS similarity	PAS (triples)	
Li (2015)	Scoring	Relatedness	BSU	
Liu et al. (2015)	Scoring	ILP	AMR	
Sherry and Bhatia (2015)	Scoring	TF-IDF	UNL	Sornlertlamvanich et al. (2001)
Subramaniam and Dalal (2015)	Rule-based	Pruning/merging through rules	RSG	Moawad and Aref (2012)
Khan et al. (2016)	Scoring	PAS similarity	PAS (graph)	Khan et al. (2015)
Li et al. (2016)	Scoring	ILP	ESLN	Li (2015)
Dohare et al. (2018)	Scoring	TF-IDF	AMR	
Khan et al. (2018)	Scoring	PageRank	PAS (graph)	
Liao et al. (2018)	Scoring	ILP	AMR	Liu et al. (2015)
Huang et al. (2020)	Machine Learning	Neural Networks	OpenIE	
Inácio and Pardo (2021)	Scoring and Machine Learning	Combined score, Random Forest	AMR	Dohare et al. (2018)

method presented better results in terms of Pyramid scoring by leveraging semantic relations.

It is important to report that not only the structure of the representations (how concepts are connected to each other) is important for summarization, but also what the actual information encoded within these structures is, as stated by Leskovec et al. (2005). The authors compared their same method with and without, so named, “linguistic features” (e.g. semantic role labels, part of speech tags, gender, named entities, among others), concluding that this type of information enhances the results compared to using only the graph structure (its topology).

Regarding the evaluation of summaries, many works – Genest and Lapalme (2011), Li (2015), Li et al. (2016), Khan et al. (2018) and Huang et al. (2020) – state that the well-known ROUGE score may not be suitable for abstractive summarization, since it relies on n-gram matching and abstracts may contain a larger variety of linguistic constructions. In their work, Huang et al. (2020), compared human annotation of errors in abstracts with their ROUGE scores and concluded that there is little to no correlation between these pieces of information, indicating that ROUGE cannot incorporate satisfactorily an error analysis as an evaluation metric. This has also been discussed by Ermakova et al. (2019) in their survey on evaluation metrics for Automatic Summarization.

Liao et al. (2018) states that their method, which uses explicit semantic representations, may be considered ‘more abstractive’ when compared to other methods which don’t. They show that their approach produces less n-grams copied directly from the input, which produces abstracts that are more similar to those created by humans. They also indicate that it seems to be easier to find relevant concepts rather than important relations.

There seems to be a consensus about how the representation quality affects the final result. Liu et al. (2015), as well as Li et al. (2016) and Inácio and Pardo (2021), show that their results obtained from gold human-annotated representations outperform the ones achieved by using automatically parsed data. This same phenomenon has been reported by Dohare et al. (2018) regarding automatic and manual coreference resolution.

It is also important to cite that Vanderwende et al. (2004) and Genest and Lapalme (2011) bring attention to how text generation methods can impact results in terms of ROUGE and linguistic quality.

This may indicate one of the many branches for further work to improve abstractive deep summarization.

5.1 Perspectives for the future

As discussed previously, an important field that can be expanded within Automatic Summarization is the one dealing with automatic evaluation of abstracts. It is known that the ROUGE score, as well as other word overlapping metrics, privileges extractive methods, as they are more homogeneous due to the fact that they copy text passages directly from the input (Ermakova et al., 2019; Huang et al., 2020). Thus, it is important to explore other automatic evaluation scores that may be more suitable in those cases.

It may also be interesting to take into account the observations of Liao et al. (2018) about relevant concepts being easier to find when compared with relations. It may be worth putting more effort into finding essential relations rather than concepts, which may improve results obtained.

As reported earlier, many works argue that the representation quality is important to ensure good results (Liu et al., 2015; Li et al., 2016; Dohare et al., 2018; Inácio and Pardo, 2021). So there must be works focused on semantic parsing and other base tools, even though they are not related specifically to Automatic Summarization. This same logic can be applied to the Natural Language Generation area, since better text generation approaches are crucial to ensure summary quality.

Acknowledgements

This research was carried out with the support of Coordination for the Improvement of Higher Education Personnel (CAPES #88887.355305/2019-00). We also had the support of the OPINANDO project[2] (PRP #668) and the POeTiSA project[3] of the Center for Artificial Intelligence (C4AI)[4] (IBM and FAPESP #2019/07665-4).

[2] https://sites.google.com/icmc.usp.br/opinando/
[3] https://sites.google.com/icmc.usp.br/poetisa
[4] https://c4ai.inova.usp.br/

References

Omri Abend and Ari Rappoport. 2017. The State of the Art in Semantic Representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89, Vancouver, Canada, July. Association for Computational Linguistics.

Virgínia B. B. Abrahão. 2018. *Semântica, enunciação e ensino*. EDUFES, Vitória, first edition.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In Antonio Pareja-Lora, Maria Liakata, and Stefanie Dipper, editors, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Boardhead, and Oren Etzioni. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-Sequence Learning using Gated Graph Neural Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia, July. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July. Association for Computational Linguistics.

Patrick Blackburn and Johannes Bos. 2005. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. Studies in Computational Linguistics. Center for the Study of Language and Information, Stanford, Calif.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Chris Brockett, Takako Aikawa, Anthony Aue, Arul Menezes, Chris Quirk, and Hisami Suzuki. 2002. English-Japanese Example-Based Machine Translation Using Abstract Linguistic Representations. In *COLING-02: Machine Translation in Asia*.

Harry Bunt, Reinhard Muskens, Gennaro Chierchia, Pauline Jacobson, and Francis J. Pelletier, editors. 1999. *Computing Meaning*, volume 73 of *Studies in Linguistics and Philosophy*. Springer Netherlands, Dordrecht.

Shu Cai and Kevin Knight. 2013. Smatch: An evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 748–752.

Deng Cai and Wai Lam. 2020. AMR Parsing via Graph-Sequence Iterative Inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online, July. Association for Computational Linguistics.

Richard Campbell and Hisami Suzuki. 2002. Language-Neutral Syntax : An Overview. Technical report, Microsoft Research, Redmond, WA.

Matheus Chaud and Ariani Di Felippo. 2018. Exploring content selection strategies for Multilingual Multi-Document Summarization based on the Universal Network Language (UNL). *Revista de Estudos da Linguagem*, 26(1):45–71.

Gennaro Chierchia. 2003. *Semântica*. Editora da UNICAMP.

Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2010. A discourse-aware graph-based content-selection framework. In *Proceedings of the 6th International Natural Language Generation Conference*.

Shibhansh Dohare, Vivek Gupta, and Harish Karnick. 2018. Unsupervised Semantic Abstractive Summarization. *Proceedings of ACL 2018, Student Research Workshop*, pages 74–83.

Liana Ermakova, Jean-Valère Cossu, and Josiane Mothe. 2019. A survey on evaluation of summarization methods. *Inf. Process. Manag.*, 56(5):1794–1814.

Ibrahim Fathy, Dalia Fadl, and Mostafa Aref. 2012. Rich semantic representation based approach for text generation. In *2012 8th International Conference on Informatics and Systems (INFOS)*, pages NLP–20–NLP–28, Cairo, May. IEEE.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, Mass.

Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from Abstract Meaning Representation using Tree Transducers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California, June. Association for Computational Linguistics.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Manuela M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, Hyderabad, India.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93, Athens, Greece, March. Association for Computational Linguistics.

Pierre-Etienne Genest and Guy Lapalme. 2011. Framework for abstractive summarization using text-to-text generation. In *Proceedings of the Workshop on Monolingual Text-to-Text Generation*, pages 64–73, Portland, Oregon. Association for Computational Linguistics.

Charles F Greenbacker. 2011. Towards a Framework for Abstractive Summarization of Multimodal Documents. In *Proceedings of the ACL 2011 Student Session*, HLT-SS '11, pages 75–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online, July. Association for Computational Linguistics.

Marcio Lima Inácio and Thiago Alexandre Salgueiro Pardo. 2021. Semantic-Based Opinion Summarization. In *Proceedings of the Conference Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications*, pages 619–628, Online. INCOMA Ltd. Shoumen, BULGARIA.

Fuad Issa, Marco Damonte, Shay B. Cohen, Xiaohui Yan, and Yi Chang. 2018. Abstract Meaning Representation for Paraphrase Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 442–452, New Orleans, Louisiana, June. Association for Computational Linguistics.

Jay J. Jiang and David W. Conrath. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, pages 19–33, Taipei, Taiwan, August. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Daniel Jurafsky and James H. Martin. 2019. Vector Semantics and Embeddings. In *Speech and Language Processing*. Prentice Hall, Upper Saddle River, third edition, October.

Atif Khan, Naomie Salim, and Yogan Jaya Kumar. 2015. A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing*, 30:737–747, May.

Atif Khan, Naomie Salim, and Haleem Farman. 2016. Clustered genetic semantic graph approach for multi-document abstractive summarization. In *2016 International Conference on Intelligent Systems Engineering (ICISE)*, pages 63–70, January.

Atif Khan, Naomie Salim, Haleem Farman, Murad Khan, Bilal Jan, Awais Ahmad, Imran Ahmed, and Anand Paul. 2018. Abstractive text summarization based on improved semantic graph approach. *International Journal of Parallel Programming*, 46(5):992–1016.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 1989–1993, Las Palmas. European Language Resources Association.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May.

Paul R. Kroeger. 2019. *Analyzing Meaning: An Introduction to Semantics and Pragmatics*. Language Science Press, Berlin, January.

Jure Leskovec, Marko Grobelnik, and Natasa Milic-Frayling. 2004a. Learning Semantic Graph Mapping for Document Summarization. In *Proceedings of ECML/PKDD 2004 Workshop on Knowledge Discovery and Ontologies*, pages 1–6, September.

Jurij Leskovec, Marko Grobelnik, and Natasa Milic-Frayling. 2004b. Learning Sub-structures of Document Semantic Graphs for Document Summarization. In *KDD2004 Workshop on Link Analysis*, Seattle, WA, USA, August.

Jure Leskovec, Natasa Milic-Frayling, and Marko Grobelnik. 2005. *Extracting Summary Sentences Based on the Document Semantic Graph*. Microsoft Research.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *26th International Conference on Computational Linguistics*, pages 236–246. Association for Computational Linguistics.

Wei Li. 2015. Abstractive multi-document summarization with semantic information extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1908–1913, Lisbon, Portugal. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for Multi-Document Summarization. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward Abstractive Summarization Using Semantic Representations. In Rada Mihalcea, Joyce Chai, and Anoop Sarkar, editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

S Mangairkarasi and S Gunasundari. 2012. Semantic based Text Summarization using Universal Networking Language. *International Journal of Applied Information Systems*, 3(8):18–23.

Inderjeet Mani. 2001. *Automatic Summarization*, volume 3 of *Natural Language Processing*. John Benjamins Publishing Company, Amsterdam.

Camilla Brandel Martins and Lucia Helena Machado Rino. 2001. Pruning UNL texts for Summarizing Purposes. In *NLPRS*, pages 539–544.

Camilla Brandel Martins and Lucia Helena Machado Rino. 2002a. Heurísticas de poda de sentenças para a sumarização automática de textos UNL. Technical Report NILC-TR-02-01, Núcleo Interinstitucional de Linguística Computacional, São Carlos, SP, Brazil, March.

Camilla Brandel Martins and Lucia Helena Machado Rino. 2002b. Revisiting UNLSumm : Improvement Through a Case Study. In *Proceedings of the Workshop on Multilingual Information Access and Natural Language Processing*.

David D. McDonald. 1992. An efficient chart-based algorithm for partial-parsing of unrestricted texts. In *Proceedings of the Third Conference on Applied Natural Language Processing -*, page 193, Trento, Italy. Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings, the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, pages 775–780, Boston, Massachusetts, USA. AAAI Press.

Ibrahim F. Moawad and Mostafa Aref. 2012. Semantic graph reduction approach for abstractive Text Summarization. In *2012 Seventh International Conference on Computer Engineering Systems (ICCES)*, pages 132–138, November.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Comput. Surv.*, 41(2), February.

Deise Cristina de Moraes Pinto, Fábio André Cardoso Coelho, Mônica Paula de Lima Cabral, and Roza Maria Palomanes Ribeiro. 2016. *Introdução à Semântica*. Fundação Cecierj, Rio de Janeiro.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical Event Extraction using Abstract Meaning Representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,, August. Association for Computational Linguistics.

Delia Rusu, Blaž Fortuna, Marko Grobelnik, and Dunja Mladeniæ. 2009. Semantic graphs derived from triplets with application in document summarization. *Informatica*, 33(3):357–362.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):21.

Sherry and Parteek Bhatia. 2015. Multilingual text summarization with UNL. In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 740–745, Ghaziabad, India, March. IEEE.

Virach Sornlertlamvanich, Tanapong Potipiti, and Thatsanee Charoenporn. 2001. UNL Document Summarization. In *Proceedings of the I International Workshop on Multimedia Annotation (MMA'2001)*, Tokyo, Japan.

Lucia Specia and Lucia Helena Rino. 2002. Representação Semântica: Alguns Modelos Ilustrativos. Technical Report NILC-TR-02-12, NILC - ICMC-USP, São Carlos, July.

Manjula Subramaniam and Vipul Dalal. 2015. Test model for rich semantic graph representation for Hindi text using abstractive method. *International Research Journal of Engineering and Technology (IRJET)*, 2(2).

Mutsuko Tomokiyo and Christian Boitet. 2016. Corpus and dictionary development for classifiers/quantifiers towards a French-Japanese machine translation. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 185–192, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Hiroshi Uchida, Meiying Zhu, and Tarcisio Della Senta. 1999. *A Gift for a Millenium*. United Nations University, Tóquio, November.

Lucy Vanderwende, Michele Banko, and Arul Menezes. 2004. Event-centric summary generation. *Proceedings of the DUC 2004*, pages 76–81.

Chuan Wang, Sameer Pradhan, Xiaoman Pan, Heng Ji, and Nianwen Xue. 2016. CAMR at SemEval-2016 Task 8: An Extended Transition-based AMR Parser. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1173–1178, San Diego, California, June. Association for Computational Linguistics.