A Short Review of Abstract Meaning Representation Applications

Nasim Tohidi[1], Chitra Dadkhah[1*]

***Abstract*-- Abstract Meaning Representation (AMR) is a representation model in which AMRs are rooted and labeled graphs that capture semantics on the sentence level while abstracting away from Morpho-Syntactic properties. The nodes of the graph represent meaning concepts and the edge labels show relationships between them. The application of AMR, as a principal form of structured sentence semantics, in Natural Language Processing (NLP) tasks is widely increasing, and it is considered a turning point for NLP research. The present study gives a brief review of the existing AMR applications in various NLP tasks. Moreover, they are compared and some of their basic features are discussed.**

***Index Terms*--Abstract Meaning Representation, Application, Natural Language Processing, Text, Semantic**

I. INTRODUCTION

All humans can answer the question Who did what whom? easily, in each context, however, it is complex for a machine to analyze it in natural language. It is about 2 decades or more, that NLP analysis relied completely on syntactic Treebanks Corpora to make machines get the meaning of human natural languages. When the Penn Treebank project [1] released the first large-scale Treebank, even, more syntactic Treebanks have been proposed for a wide range of languages. Then, they have been used to build principal NLP systems, such as Part-Of-Speech (POS) taggers, Machine Translation (MT) systems, and Question Answering (QA) [2-6].

By passing from the syntactic structure analysis to semantics, scientists found statistical parsers not well suited for meaning representation production. In semantic analysis, complicated structures, which are difficult to capture by parse tree structures and their limitations have often been encountered. For instance, in a semantic network, nodes are often equivalent to the argument of more than one predicate. So, it can be useful for finding semantically less important words, hence, leaving nodes, that do not add any further meaning to the final result, unattached. To solve the problems posed by this limitation and do a direct semantic analysis of all sentences, recent research has shifted to parsing with graph-structured representations. Because syntactic Treebanks had been vital for enhancing the performance of syntactic parsers, emerge techniques with semantic parsing using Sembanks, which are sets of English sentences paired with their related semantic representations [7].

Banarescu et al. in [8] tried to annotate the logical meaning of sentences in Abstract Meaning Representation (AMR), which constituted semantic roles, questions, co-reference, modality, negation, and linguistic phenomena. Thus, by producing a notable corpus and a correctable logical semantic input format, the AMR creators hope to be able to encourage important advances in Statistical Machine Translation (SMT), Natural Language Generation (NLG), and Statistical Natural Language Understanding (SNLU).

In this paper, some of the main AMR applications in NLP tasks are studied and compared based on the relevant papers reviewed. The rest of this paper is structured as follows: Section II investigates AMR briefly. Section III gives an overview of AMR applications in various NLP tasks. Finally, Section IV provides the conclusion and some outlooks for future research.

II. ABSTRACT MEANING REPRESENTATION

AMRs are commonly considered tree structures; however, they can be seen as directed acyclic graphs with a single root, where vertices are variables and edges denote roles and instances. As a result, AMRs can be converted into sets of triples [9]. The tree structure is more useful for semantic interpretation since we must be able to determine the scope for operators like negation.

It is very simple to provide a semantic and theoretical interpretation. AMR can be made just by converting roles into two-place predicates, concepts, and events into one-place predicates, and by quantifying the existence of all variables introduced by events and concepts. Furthermore, it is noteworthy that this kind of representation does not allow us to systematically include scope-based operators, such as quantification, negation, and projection.

Moreover, a formal definition of AMRs syntax can be provided and a recursive translation function from AMR to FOL (First Order Logic) can be produced. The function, which has many similarities with the conversion from AMR to λ-calculus, is proposed in [10].

The produced structure is a closed formula, meaning that all of its variables are bounded because the translation certifies that no free occurrences of variables can be

1: N. Tohidi is with the Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran

Corresponding author: dadkhah@kntu.ac.ir

revealed. In addition, interestingly, simple AMRs, which are very similar to the controlled DRT[1] fragment, are presented in [11]. Simple AMRs are in the two-variable fragment of FOL. It should be noted that FOL is not decidable. In contrast, the two-variable fragment is a decidable FOL, in which formulas have a maximum of two variables with different names; however, it does not have function symbols, yet probably has equality. In addition, it has the property of a finite model, that is, if a fragment formula is satisfied, it can also be satisfied in a finite model [12].

Generally, AMR is constructed based on the following points [13]:

- **Graph Representation**: As mentioned earlier, AMRs are rooted, labeled, directed graphs that allow co-references to be modeled by reentrancy. To represent the human reading and writing structure, the AMR format uses the PENMAN notation [14, 15].
- **Abstraction:** We know that AMRs abstract away from morphological and syntactic diversity. Therefore, different sentences could have the same AMR, if they exactly have the same semantics, even with different structures. Besides, this results in the following principle: in the annotation, no particular alignment between graph components and string has been provided.
- **Framesets:** AMRs predicates are annotated based on framesets specified in Propbank [16].

Fig. 1 illustrates an example of AMR annotation of the sentence Private rights must conform to the public welfare. Usually, nodes are recognized with their variable. For instance, *c* is labeled with the concept conform-01. Moreover, the labeled edges connecting nodes are relations, such as ARG1. Plus, nodes with no variables are constants, which are usually used to represent name, negation, or number.

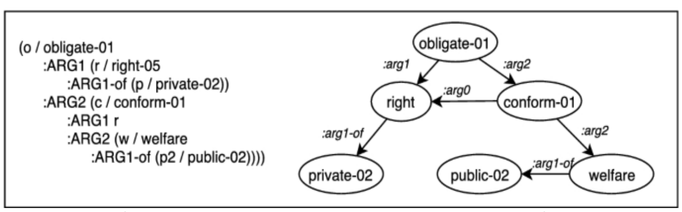

Fig. 1. AMR for an example sentence in PENMAN and graph format [17].

In most cases, AMR concepts can be related to a single word in the sentence constituting a one-to-one mapping. However, sometimes, there are concepts, which cannot easily be associated with any specific word in the sentence. These concepts usually indicate inferred knowledge, which is invoked by certain phrases or implicit relationships between disparate clauses. This type of concept is called Abstract Concept. For instance, the concept country can be an inferred named entity type for Japan.

It should be noted that most of the relevant research was done in the English language, and more effort is needed for other languages. One of the reasons for less research in non-English languages could be its structure. AMR was designed specifically for English, and it may not be easy to use the whole format in some languages.

[1] Discourse Representation Theory

III. AMR APPLICATIONS

The existence of AMR tools and their quality performance has encouraged many scientists to work on integrating the whole sentence meaning into NLP functions [18]. Table I represents the distribution of the studied different applications of AMR in the present study.

TABLE I
The Distribution of the Different Applications Of AMR In NLP Tasks.

Applications	Percent
Machine Comprehension	31 %
Text Summarization	18 %
Question Answering	18 %
Entity Linking and Linked Data	13 %
Machine Translation	9 %
Information Retrieval	9 %
Other	2 %
Total	100 %

As seen in Table I, AMR has been used in different NLP tasks. In the coming paragraphs, some examples of main research applications are studied. It is necessary to note, here, the purpose is to show the usefulness of this representation method. That is why, only some examples of each application have been discussed, however, the number of research in each field is more than this number.

A. *Text Summarization*

In the text summarization process, the size of the original documents would be reduced so that they become a much briefer text, and the important facts in the original documents are retained [19].

Liu et al., in [20], introduced an abstractive summarization framework, which its purpose was not limited to building extractive and compressive summarization. In their work, each sentence was parsed to reach separate AMR graphs. Then, these graphs were merged to make a fully dense graph, from which a summary graph was created by choosing a subset of nodes and edges. So, they had a heuristic generator whose input was a summary graph and it was developed to build summarized texts. They applied the graph-to-graph transformation method to reduce the source semantic graph to a summary graph by using an existing AMR parser and assuming the availability of an AMR-to-text generator. Their framework was data-driven and was not designed for a particular domain.

Hardy and Vlachos, in [21], worked on abstractive summarization using AMR with a neural language generation stage, which was guided using the source document. They represented that the guidance process could improve summarization results. Besides, they found that the overall summarization performance on later parses was higher than the neural encoder-decoder technique trained on a larger dataset.

Mishra and Gayen, in [22], defined the concept of lossless summary. Their proposed approach aimed to create automatic summaries without any loss of information by removing the dangling anaphora, that resulted in an incoherent summary. Their research aimed to solve the problem of incoherency in extractive summaries. They introduced a pipeline of operations, as shown in Fig. 2, in order to generate summaries. In their research, a co-reference resolution was performed pairwise on sentences before generating AMR of them. Then, they developed an algorithm to combine AMR graphs. At last, the text was generated using the combined AMR graphs. As seen in Fig. 2, the proposed approach aimed to create automatic summaries without any loss of information. In this regard, a more coherent lossless summary from the given text was produced by removing the dangling anaphora, that resulted in an incoherent summary. AMR of the co-referenced news article was merged using the author's proposed graph merging algorithm.

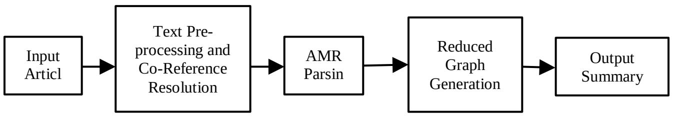

Fig. 2. The pipeline for the approach proposed in [22].

B. Question Answering

The main goal of a question-answering system is to communicate with humans via natural language, directly. So, they get the user's questions in natural language and give accurate responses. Hence, such systems must deal with complex NLP methods, and some researchers tried to reach this goal by using AMRs.

Mitra and Baral, in [23], designed a model of an agent, which used the Answer Set Programming (ASP) language as the primary knowledge representation, and reasoning language, along with the standard statistical NLP models. Given a training dataset, which included a set of narrations, questions, and their answers, and an inductive logic programming algorithm, the agent jointly used a translation system, and statistical NLP methods to learn the knowledge required for answering similar questions. In their implementation, the agent model contained three layers, two of which (statistical inference and translation layers) have used AMR parser to achieve their goal as follows:

- *Statistical Inference Layer* comprised statistical NLP models, however, in their case study, it included just an AMR.
- *The formal Reasoning Layer* was responsible for formal reasoning. It applied the ASP language as the reasoning language and knowledge representation. The knowledge needed for reasoning was learned using a modified version of the inductive logic programming algorithm XHAIL. The reasoning module took sentences represented in the logical language of Event calculus, which is a temporal logic for reasoning about the events and their efforts. The ontology of the Event calculus contained time points, fluent (i.e., properties that have certain values in time), and event (i.e., occurrences in time, which may affect fluent and change their value). Plus, the formalism included 2 domain-independent axioms (the last line in Table 2) to combine the commonsense law of inertia, according to which fluent persists over time unless it was affected by an event. The building blocks of event calculus and its domain-independent axioms are illustrated in Table II.
- *The translation layer* encoded the natural language sentences to the syntax of event calculus by applying the AMR parser from the statistical inference layer. This layer communicated with both other layers and let information be passed from one layer to another. In their case study, they applied a naive deterministic algorithm to form the translation layer.

TABLE II
The Principal Predicates and Axioms of Simple Discrete Event Calculus (SDEC) [23].

Predicate	Meaning
happensAt(F, T)	Event E occurs at time T
initiatedAt(F, T)	At time T, a period of time for which fluent F holds is initiated
terminatedAt(F, T)	At time T, a period of time for which fluent F holds is terminated
holdsAt(F, T)	Fluent F holds at time T
holdsAt(F, T + 1) ← initiatedAt(F, T)	holdsAt(F, T + 1) ← holdsAt(F, T), not terminatedAt(F, T)

Michael et al., in [24], proposed Question-Answer Meaning Representations (QAMRs), that represented the predicate-argument structure of a sentence as a set of question-answer pairs. They created a crowdsourcing scheme to represent that QAMRs could be labeled by a slight training phase. They collected a dataset including over 5,000 sentences and 100,000 questions. Analyzing the quality showed the crowd-generated question-answer pairs covered the vast majority of predicate-argument relationships in existing datasets, like PropBank [16], NomBank [25], and QA-SRL [26], besides many previously under-resourced ones such as implicit arguments and relations.

Bonial et al., in [27], applied AMR for recognizing answers to research questions in medical scientific documents, specifically, to boost the study of UV (Ultra-Violet) inactivation of the coronavirus causing the disease COVID-19. They explained the development of a proof-of-concept prototype tool, called Info-Forager, that exploits AMR to do a semantic search, aiming at meaning the user question, and matching it to sentences in medical documents possibly including information to answer that question. Their research determined an opportunity for NLP tools to help in automatically sieving through a large number of documents to detect related answers to particular and aimed questions of relevant experts working in the field of UV inactivation of viruses. They proposed Info-Forager (Fig.3), a proof-of-concept prototype tool used by applying semantic understanding and search, that can go beyond the lexicons in a user question to concentrate on its meaning. By exploiting a semantic search, they assumed the user can more simply search via medical documents, as they did not need to re-phrase their questions to adapt to the system's limitations. Info-Forager first parsed a user research question into AMR, then compared the resulting AMR query to a collection of medical research papers that already were parsed into AMR. All query-sentence pairs of AMRs in each paper were scored for their meaning similarity, and the model returned the highest-ranking answer sentence and the source document.

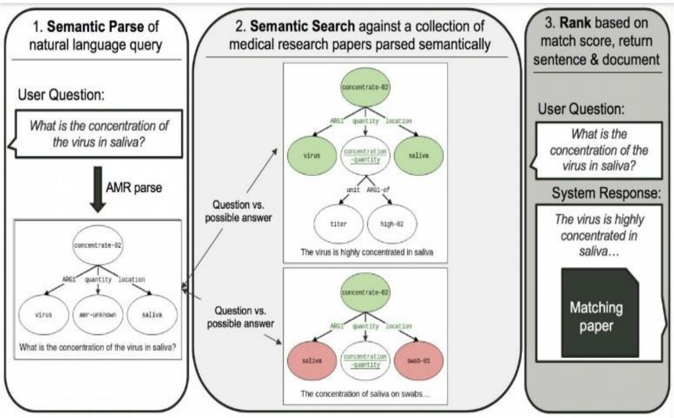

Fig. 3. Info-Forager overview [27].

As depicted in Fig. 3,

1. The user question was automatically parsed into AMR (parsed query).
2. Parsed-query was compared to a collection of research papers already parsed into AMR (parsed-answers).
3. Matches were ranked and the highest-ranking sentence was returned with its source document.

In Fig. 4, the whole process is presented in 8 steps.

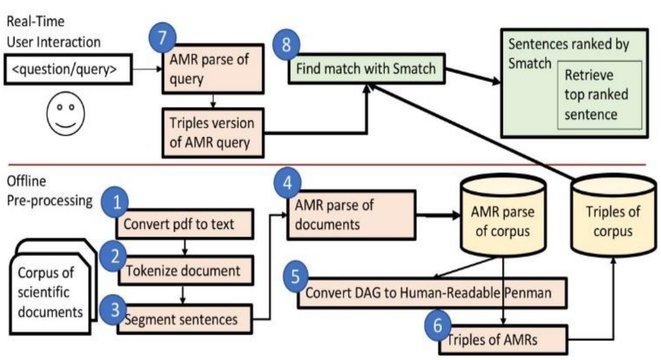

Fig. 4. Info-Forager Prototype Pipeline [27].

Kapanipathi et al., in [28], designed a Neuro-Symbolic QA (NSQA) model, a modular Knowledge Base QA (KBQA) system, that exploited:

1. Parses of AMR for question understanding in a task-independent manner.
2. An efficient graph transformation strategy for converting AMR parses to candidate logical queries, which were aligned to the knowledge base.
3. A pipeline-based method that combined several reusable modules trained particularly for their individual tasks (entity and relationship linkers, semantic parser, and neuro-symbolic reasoner). Besides, it did not need end-to-end training data.

Their proposed method delegated the complication of NLU (questions) to AMR parsers, diminished the demand for text-to-SPARQL (end-to-end) training data by a pipeline structure, in which each module was trained for its particular sub-task, and simplified the use of an independent reasoner by applying an intermediate logic form. Table III shows the different question types supported by NSQA with an example for each type.

TABLE III
Various Question Types Supported by NSQA [28].

Question Type	Example	Supported
Simple	Who is the mayor of New York?	Yes
Multi-relational	Give me all actors starring in movies directed by Christopher Nolan.	Yes
Count-based	How many theories did Albert Einstein come up with?	Yes
Superlative	What is the highest mountain in Asia?	Yes
Comparative	Does Money Heist have more episodes than Breaking Bad?	No
Geographic	Was Tom Cruise born in the US?	Yes
Temporal	When the final match of the football world cup 2022 will start?	No

Fig. 5 illustrates the pipeline of the NSQA model for an example input question (Which actors starred in Spanish movies produced by Benicio del Toro?) in natural language. The main steps were as follows:

1. Parsing questions to the related AMR graphs.
2. Transforming the graph to a set of candidate KB-aligned logical queries, using a naive graph transformation method.
3. Applying a Logical Neural Network (LNN) in order to reason over KB facts, and producing answers to the queries.

In Fig. 5, the representation for the two unknown variables across all steps are underlined: AMR-aligned tokens (Which, movies), AMR graph (amr-unknown, movie), paths representation (amr-unknown, movie), logical representation (a as actor, m as movie) and SPARQL interpretation (?actor, ?movie). Additionally, AMR, Entity Linking, and Relation Linking outputs are shown in green, blue, and orange, respectively.

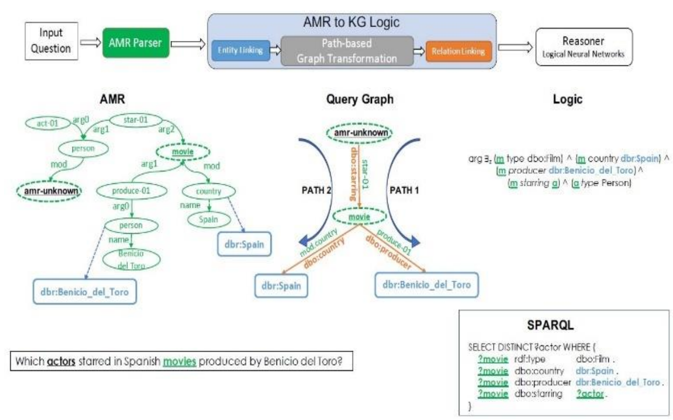

Fig. 5. The pipeline of the NSQA model for an example input question [28].

It can be concluded that applying semantic parses, like AMR, in comparison with syntactic dependency parses, resulted in some benefits for KBQA systems as follow:

1. Independent advances in AMR parsing, which were beneficial from various aspects, could enhance the general performance of the system.
2. AMR offered a normalized form of input questions, which made NSQA resistant to delicate changes in input questions with the same meaning.
3. AMR could represent sentences with complicated structures, like imperative statements or multi-hop questions, transparently.

Nonetheless, it is undeniable that the application of AMR semantic parses in NSQA has some challenges as Error propagation, Granularity, and mismatch Optimization.

1. Error propagation: AMR errors can be propagated in the pipeline and lead to errors in producing the expected answer.
2. Granularity mismatch: Their introduced path-based AMR transformation was general (without any domain-specific stimulus). Thus, the algorithm required further modifications in new domains, considering the different granularity between SPARQL and AMR.
3. Optimization mismatch: The optimization metric for AMR training (Smatch) was sub-optimal for KBQA; NSQA needed a specific subset of paths to be properly extracted because Smatch equally focused on all edge-node triples.

Deng et al., in [29], designed a Question Decomposition method based on AMR (QDAMR) for multi-hop QA, that achieved interpretable reasoning by decomposing a multi-hop question into simpler sub-questions and answering them in order. As annotating the decomposition is expensive, first they delegated the complexity of understanding the multi-hop question to an AMR parser. Afterward, they achieved the decomposition of a multi-hop question via the segmentation of the corresponding AMR graph based on the required reasoning type. Eventually, they generated sub-questions by using an AMR-to-Text generation model and answered them with an off-the-shelf QA model. These steps are shown in Fig. 6.

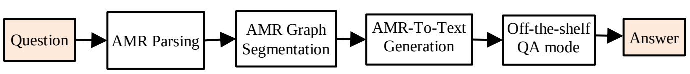

Fig. 6. An overview of the QDAMR framework for multi-hop QA [29].

C. *Information Retrieval*

Information Retrieval (IR) is the activity of obtaining information system resources, that are relevant to an information need of the user, from a collection of those resources.

Garg et al., in [30], used the effectiveness of applying semantic graphs in bio-molecular interaction extraction. Given the parsed AMR graphs of the biomedical text, they presented a graph-kernel-based algorithm to score candidate interactions. This algorithm considerably outperforms a baseline system, which relies on syntax-based features. They designed and implemented a model to extract bio-molecular interactions, which applied deep semantic parses AMRs of biomedical texts. Plus, they proposed a method relying on GDK[2] for extracting document-level interactions from an AMRs set. GDK could be applied jointly on both AMR and SDG[3] parses of sentences because while neither parsing method is thorough, their composition can yield premier outcomes. To simplify the joint method, they defined an edge vector space embedding model to evaluate the similarity level among various parse types.

One of the most important issues in IR is paraphrase detection. Issa et al., in [31], illustrated that the naïve use of AMR in paraphrase detection is not useful necessarily. So, they tried to define a method based on latent semantic analysis in combination with AMR parsing. They described an approach to incorporate an AMR parser output into the detection of paraphrases. More precisely, their proposed model combined two graphs required to be tested for a paraphrase relation, and then, re-weighted a sentence-term matrix by the PageRank [32] values of the vertexes in the combined graph.

D. *Entity Linking and Linked Data*

In NLP, Entity Linking (EL), referred to Named-Entity Linking (NEL), Named-Entity Disambiguation (NED), Named-Entity Recognition and Disambiguation (NERD), or Named-Entity Normalization (NEN), which is the task of assigning a unique identity to entities, such as locations, famous individuals, so on, that is mentioned in a text.

Pan et al., in [33], applied AMR for the EL task. They believed that EL needs a representation for relations among entities in texts. Their system used the AMR graph as a rich semantic context for a given entity. They combined it with an unsupervised graph inference algorithm, so it outperformed EL systems, that rely on Semantic Role Labeling (SRL) information. They proved that AMR can represent the contexts of entity mentions for EL more efficiently in comparison with previous methods. Besides, according to Table IV, they demonstrated that AMR enables EL performance comparable to the supervised methods by applying an unsupervised, non-collective strategy.

TABLE IV
Accuracy (%) on a Test Set With 1613 Mentions [33].

Approach		News documents	Discussion forum posts	Total
Popularity	Commonness	89.76	68.99	82.20
	Google Search	88.10	77.17	84.12
Supervised	State-of-the-art	93.07	87.41	91.01
Unsupervised Context Collaborator Approach	Sentence-Level Co-occurrence	93.17	73.25	85.92
	Document-Level Co-occurrence	90.05	69.86	82.69
	Human AMR	93.56	86.88	91.13
	System AMR	90.15	85.69	88.52
	Human SRL	93.27	71.21	85.24
Unsupervised Combined Approach	Human AMR	94.34	88.25	92.12

Huang et al., in [34], presented a modern unsupervised entity-typing framework by combining distributional and symbolic semantics. They started by learning three types of representations for each entity: general meaning representation, specific context representation, and knowledge representation based on knowledge bases. Next, they designed a joint hierarchical clustering and linking algorithm in order to type all mentions using these representations. Their framework did not rely on any annotated data, handcrafted features, or pre-defined typing schema; so, it could be adapted to a new domain, genre, or

[2] Graph Distribution Kernels

[3] Stanford Dependency Graphs

language, quickly. They evaluated the results in several languages, including English, Japanese, Chinese, Hausa, Yoruba, and some general and biomedical domains, and proved the portability of their framework.

Burns et al., in [35], introduced an AMR corpus as Linked Data (AMR-LD) and the methods used to produce it, such as an open-source implementation. It has many advantages, like convenient analysis using SPARQL queries, ontology inferences enabled by embedding in the web of LD, and the impact of semantic web representations directly derived from natural language.

Zhang and Ji, in [36], introduced an AMR-guided framework for joint Information Extraction (IE) in order to discover events, entities, and relations by applying a pre-trained AMR parser. It contained two main components as follows:

1. An AMR-based semantic graph aggregator; enabled the candidate entity and event trigger nodes to collect neighborhood information from the AMR graph in order to transfer messages between related knowledge elements.
2. An AMR-guided graph decoder; was used to extract knowledge elements, according to the order established by the hierarchical structures in AMR.

According to Table V, their experiments on multiple datasets showed that the AMR graph encoder and decoder have provided significant advantages in comparison with other related works.

TABLE V
F-Scores (%) On Dev Set for Joint Information Extraction on Bionlp Genia Datasets [36].

Dataset	Model	Task			
		Entity	Event Trigger	Event Argument	Relation
Genia'11	OneIE	81.8	56.9	57.0	63.1
	AMR-IE	82.2	61.5	59.8	65.2
Genia'13	OneIE	71.5	57.3	51.4	39.3
	AMR-IE	78.4	63.8	58.0	42.4

E. *Machine Translation*

MT is a branch of computational linguistics, which studies the application of machines in translating text or speech from one language to another.

Song et al., in [37], investigated the advantages of AMR

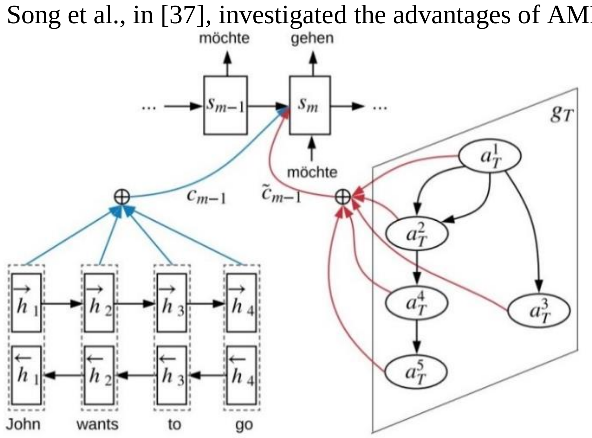

Fig. 7. Overall architecture of the model proposed in [37].

on neural MT. Their results on a standard English-to-German dataset represented that settling AMR as extra knowledge can considerably improve a strong attention-based sequence-to-sequence (Seq2seq) neural translation model. Fig. 7 illustrates the overall architecture of their model, which adopts a BiLSTM and their Graph Recurrent Network (GRN) for encoding the source sentence and AMR, respectively.

As seen in Fig. 7, an attention-based LSTM decoder was applied to generate the output sequence in the target language, with attention models over both the sequential encoder and the graph encoder.

Pham et al., in [38], designed an extension of the convolutional neural MT model to incorporate AMR as a kind of meaning representation to decrease language ambiguity or reduce data sparseness issues. They implemented a translation system from English to Vietnamese. To be more precise, they proposed a method utilizing external knowledge (AMR graphs) to enhance translation quality. Fig. 8 shows the overall architecture of their graph encoder for encoding AMR graphs and their improved decoder to incorporate graph knowledge to enrich deep Convolutional Neural Network (CNN) representations.

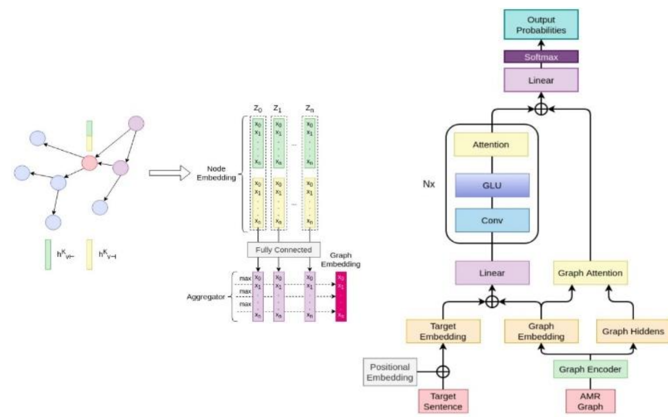

Fig. 8. Overall architecture of the graph encoder (left) and decoder (right) proposed in [38].

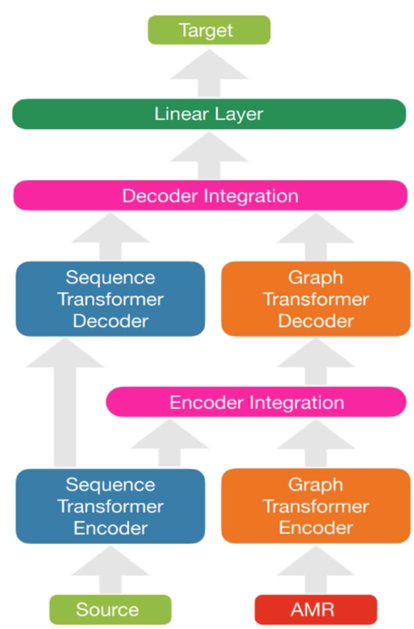

Fig. 9. Overview of the AMR-Transformer model [39].

As depicted in Fig. 8, one more attention mechanism was used because there is no one-to-one correspondence mapping between AMR nodes and words in a sentence.

Li, Jeffrey Flanigan, in [39], introduced an encoder-decoder model augmenting the Transformer-based model with a Heterogeneous Graph Transformer (HGT), which encodes source sentence AMR graphs. Fig. 9 illustrates the overview of their proposed model architecture.

As shown in Fig. 9, to encode and decode both source sentences and source AMR graphs to target sentences, their model included two parallel stacked encoder and decoder layers, one for sequence encoding and decoding from the neural Seq2seq model, and the other for graph encoding and decoding from the neural graph to the sequence model. Given the encoded sequence representation from the sequence encoder and the encoded graph representation from the graph encoder, the Seq2seq decoder only received the sequence representation while the graph-to-sequence decoder receives the combination of the sequence representation and the graph representation. Eventually, 2 decoder representations were concatenated and fed into the final linear layer to generate a target sequence representation. In this respect, the model could combine the merits of the traditional Seq2seq model, which does translation on source sentence encodings, and the graph-to-sequence model, which incorporates AMR graphs into the translation. The combination of source sentence representation and the graph representation in the graph-to-sequence decoder could lead the graph-to-sequence decoder to decode towards good translation quality since using only AMR graphs representation could result in poor translation quality compared to the vanilla Seq2seq model using source sentences.

F. *Machine Comprehension*

The main reason for generating the semantic bank is to achieve the significant NLP goal, which is NLU. Usually, machine comprehension task focuses on one of the facets of NLU to test the ability of a machine to understand and reason with natural language.

Sachan and Xing, in [40], presented a machine comprehension system integrated by AMR. Like the method in [20], a graph representation for the passage and question was built by creating the interactions among parsed AMR graphs. Then, by modeling both sub-graph selection and question mapping with latent variables, a unified max-margin method was used to jointly learn the latent structures.

Lyu et al., in [41], proposed a neural parser considering alignments as latent variables in a joint model of probabilistic concepts, alignments, and relations. For a careful deduction, they needed to marginalize over alignments, which was infeasible. Therefore, they applied the variation auto-encoding framework, besides a continuous relaxation of the discrete alignments. Also, they proved that joint modeling was better than a pipeline of align and parse.

Liao et al., in [42], investigated the feasibility of applying AMR in the form of content representation. Their method compressed source documents into a set of summary graphs, which were in the form of AMR graphs. Then, in a surface realization step, these graphs were transformed into a set of summary sentences. Their approach was completely flexible and data-driven. In addition, each component could optimize independently by a small part of the available training data. The overall architecture of their model is illustrated in Fig. 10.

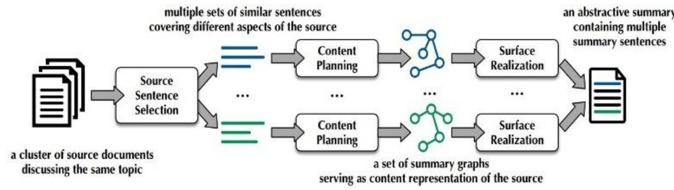

Fig. 10. Overall architecture of the model proposed in [42].

Their main purpose was to produce a text abstract containing multiple sentences from a cluster of news articles about a given topic. As shown in Fig. 10, their proposed model had three major components.

1. Source Sentence Selection
2. Content Planning
3. Surface Realization

The first component received a set of news articles as input and chose sets of similar sentences covering various aspects of the topic; the second one used a set of similar sentences and derived a summary graph from them; the last component transformed a summary graph into a sentence in natural language. Their proposed model allowed each of these three components to be optimized individually using small-scale in-domain training data and decreasing the requirement of large-scale parallel training data.

As mentioned in previous sub-sections, Song et al., in [37], worked on the efficiency of AMR for neural MT. In their study, they used a standard English-to-German dataset, and they represented that incorporating AMR as additional knowledge can notably optimize a strong attention-based Seq2seq neural translation approach.

Bonial et al., in [43], introduced a schema enriching AMR in producing a semantic representation for facilitating NLU in dialogue systems. In this schema, they explored dialogue in a human-robot interaction domain, wherein a conversational robot was employed for search and navigation tasks, together with a human partner. In addition, they expanded a list of speech acts, which were appropriate for their domain, so they proposed Dialogue-AMR. It was an improved AMR that represented the utterance content, the illocutionary force in it, and its aspect and tense. Moreover, to evaluate the model coverage, they applied manual and automatic methods, resulting in the construction of the DialAMR corpus. It was a rich corpus of annotated human-robot dialogue.

Recently, Elbasani, and Kim, in [44], proposed a data-driven approach employing AMR to extract the online text content's meaning into a CNN to determine the level of profanity. They used AMR to detect the meaning representation from the input, which was the sentence annotated as a profane or offensive phrase. Next, they applied CNN algorithms to determine whether the input is a profane phrase or not. In other words, they exploited CNN to learn the graph's local features and capture the toxic meaning of the sentence. They proved that since AMR is capable of describing the correlation of each word and expressing the correlation kind that the word has, CNN could capture the toxic meaning from the text.

IV.CONCLUSION

AMR was first presented in [8] and explained in more detail in the AMR annotation guidelines. Generally, AMR refers to rooted, directed, labeled, and acyclic graphs representing the meaning of whole sentences. Their main goal is to abstract away from syntactic representations, in a way that semantically similar sentences will be assigned the same AMR, even if they are not completely the same and identically worded. Initially, AMR was designed just for English language, although, step by step, by some modifications, it has been applied for some other natural languages, too.

In this paper, we reviewed some of the main AMR applications in primary NLP tasks from 2013-2022, in 6 different groups:

1. Text Summarization
2. Question Answering
3. Information Retrieval
4. Entity Linking and Linked Data
5. Machine Translation
6. Machine Comprehension

There are various directions for possible future AMR-based works. Specifically, further enhancing the AMR corpus in different natural languages can result in shared tasks on natural language understanding and generation. Hence, it is recommended to advance the field and drive new interests in graph-based semantic parsing and generation. Eventually, the AMR language is frequently subject to change; ultimately, it can include more relations, quantification, entity normalization, or temporal and modal relations. Furthermore, a comprehensive list of more abstract frames can be imagined.

V. REFERENCES

[1] Marcus, Mitchell P.; Santorini, Beatrice; Marcinkiewicz, Mary Ann, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics,* vol. 19, no. 2, pp. 313-330, 1993.

[2] Popel, Martin; Tomkova, Marketa; Tomek, Jakub; Kaiser, Łukasz; Uszkoreit, Jakob; Bojar, Ondřej; Žabokrtský, Zdeněk, "Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals," *Nature Communications,* vol. 11, no. 1, 2020.

[3] J.-M. Torres-Moreno, Automatic Text Summarization, Wiley-ISTE, 2014.

[4] Abolghasemi, Majid; Dadkhah, Chitra; Tohidi, Nasim, "HTS-DL: Hybrid Text Summarization System using Deep Learning," in *The 27th International Computer Conference, the Computer Society of Iran,* Tehran, Online, 2022.

[5] Tohidi, Nasim; Dadkhah, Chitra; Rustamov, Rustam B., "Optimizing Persian Multi-objective Question Answering System," *International Journal on Technical and Physical Problems of Engineering (IJTPE),* vol. 13, no. 46, pp. 62-69, 2021.

[6] Tohidi, Nasim; Hasheminejad, Seyed Mohammad Hossein, "Optimizing Question Answering Systems by Accelerated Particle Swarm Optimization (APSO)," *Signal and Data Processing,* vol. 52, no. 2, pp. 161-174, 2022.

[7] J. Flanigan, "Parsing and Generation for the Abstract Meaning Representation," Carnegie Mellon University, Pittsburgh, 2018.

[8] Banarescu, Laura; Bonial, Claire; Cai, Shu; Georgescu, Madalina; Griffitt, Kira; Hermjakob, Ulf; Knight, Kevin; Koehn, Philipp; Palmer, Martha; Schneider, Nathan, "Abstract Meaning Representation for Sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, Sofia, Bulgaria, 2013.

[9] Cai, Shu; Knight, Kevin, "Smatch: an Evaluation Metric for Semantic Feature Structures," in *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, 2013.

[10] Artzi, Yoav; Lee, Kenton; Zettlemoyer, Luke, "Broad-coverage CCG semantic parsing with AMR," in *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[11] J. Bos, "Economical Discourse Representation Theory," in *International Workshop on Controlled Natural Language*, Verlag Berlin Heidelberg, 2010.

[12] M. Mortimer, "On languages with two variables," *Mathematical Logic Quarterly,* vol. 21, no. 1, pp. 135-140, 1957.

[13] C. Wang, "Abstract Meaning Representation Parsing," Ph.D. thesis, Brandeis University, 2018.

[14] Matthiessen, Christian M. I. M.; Bateman, John A., Text Generation, and Systemic-Functional Linguistics: Experiences from English and Japanese (Communication in Artificial Intelligence), London: Pinter Pub Ltd, 1991.

[15] W. C. Mann, "An overview of the Penman text generation system," in *In Proceedings of the Third AAAI Conference on Artificial Intelligence*, 1983.

[16] Palmer, Martha; Gildea, Daniel; Kingsbury, Paul, "The Proposition Bank: An Annotated Corpus of Semantic Roles," *Computational Linguistics,* vol. 31, no. 1, pp. 71-106, 2005.

[17] Vu, Sinh Trong; Nguyen, Minh Le; Satoh, Ken, "Abstract meaning representation for legal documents: an empirical research on a human-annotated dataset," *Artificial Intelligence and Law,* vol. 30, p. 221–243, 2022.

[18] Tohidi, Nasim; Dadkhah, Chitra, "A Study on Abstract Meaning Representation Applications," in *The first Conference on Artificial Intelligence and Smart Computing*, Online, 2022.

[19] Tohidi, Nasim; Hasheminejad, Seyed Mohammad Hossein, "A Practice of Human-Machine Collaboration for Persian Text Summarization," in *The 27th International Computer Conference, the Computer Society of Iran*, Online, 2022.

[20] Liu, Fei; Flanigan, Jeffrey; Thomson, Sam; Sadeh, Norman; Smith, Noah A., "Toward Abstractive Summarization Using Semantic Representations," in *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, 2015.

[21] Hardy, Hardy; Vlachos, Andreas, "Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018.

[22] Mishra, Ritwik; Gayen, Tirthankar, "Automatic Lossless-Summarization of News Articles with Abstract Meaning Representation," *Procedia Computer Science,* vol. 135, pp. 178-185, 2018.

[23] Mitra, Arindam; Baral, Chitta, "Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[24] Michael, Julian; Stanovsky, Gabriel; He, Luheng; Dagan, Ido; Zettlemoyer, Luke, "Crowdsourcing Question-Answer Meaning

Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, 2018.

[25] Meyers, Adam; Reeves, Ruth Madeleine; Macleod, Catherine; Szekely, Rachel; Zielinska, Veronika; Young, Brian; Grishman, Ralph, "The NomBank Project: An Interim Report," New York University, 2009.

[26] He, Luheng; Lewis, Mike; Zettlemoyer, Luke, "Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language," in *The 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[27] Bonial, Claire; Lukin, Stephanie M.; Doughty, David; Hill, Steven; Voss, Clare, "InfoForager: Leveraging Semantic Search with AMR for COVID-19 Research," in *The Second International Workshop on Designing Meaning Representations*, Barcelona, Spain (online), 2020.

[28] Kapanipathi, Pavan; Abdelaziz, Ibrahim; Ravishankar, Srinivas; Roukos, Salim; Gray, Alexander; Astudillo, Ramón Fernandez; Chang, Maria; Cornelio, Cristina; Dana, Saswati; Fokoue, Achille; Garg, Dinesh; Gliozzo, Alfio; Gurajada, Sairam; Karanam, Hima, "Leveraging Abstract Meaning Representation for Knowledge Base Question Answering," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, 2021.

[29] Deng, Zhenyun; Zhu, Yonghua; Chen, Yang; Witbrock, Michael; Riddle, Patricia, "Interpretable AMR-Based Question Decomposition for Multi-hop Question Answering," in *The Thirty-First International Joint Conference on Artificial Intelligence (IJCA-22)*, Vienna, 2022.

[30] Garg, Sahil; Galstyan, Aram; Hermjakob, Ulf; Marcu, Daniel, "Extracting biomolecular interactions using semantic parsing of biomedical text," in *AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[31] Issa, Fuad; Damonte, Marco; Cohen, Shay B.; Yan, Xiaohui; Chang, Yi, "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*, New Orleans, Louisiana, 2018.

[32] Page, Lawrence; Brin, Sergey; Motwani, Rajeev; Winograd, Terry, "The PageRank Citation Ranking: Bringing Order to the Web," Technical report, Stanford InfoLab, 1999.

[33] Pan, Xiaoman; Cassidy, Taylor; Hermjakob, Ulf; Ji, Heng; Knight, Kevin, "Unsupervised Entity Linking with Abstract Meaning Representation," in *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, 2015.

[34] Huang, Lifu; May, Jonathan; Pan, Xiaoman; Ji, Heng; Ren, Xiang; Han, Jiawei; Zhao, Lin; Hendler, James A., "Liberal Entity Extraction: Rapid Construction of Fine-Grained Entity Typing Systems," *Big Data*, vol. 5, no. 1, pp. 19-31, 2017.

[35] Burns, Gully A.; Hermjakob, Ulf; Ambite, Jose Luis, "Abstract Meaning Representations as Linked Data," in *International Semantic Web Conference*, 2016.

[36] Zhang, Zixuan; x Ji, Zixuan, "Abstract Meaning Representation Guided Graph Encoding and Decoding for Joint Information Extraction," in *The 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, 2021.

[37] Song, Linfeng; Gildea, Daniel; Zhang, Yue; Wang, Zhiguo; Su, Jinsong, "Semantic Neural Machine Translation Using AMR," *Transactions of the Association for Computational Linguistics,* vol. 7, pp. 19-31, 2019.

[38] Pham, Viet; Nguyen, Long H.B.; Dinh, Dien, "Semantic Convolutional Neural Machine Translation Using AMR for English-Vietnamese," in *CCCIS 2020: Proceedings of the 2020 International Conference on Computer Communication and Information Systems*, 2020.

[39] Li, Changmao; Flanigan, Jeffrey, "Improving Neural Machine Translation with the Abstract Meaning Representation by Combining Graph and Sequence Transformers," in *The 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, Seattle, Washington, 2022.

[40] Sachan, Mrinmaya; Xing, Eric, "Machine Comprehension using Rich Semantic Representations," in *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, 2016.

[41] Lyu, Chunchuan; Titov, Ivan, "AMR Parsing as Graph Prediction with Latent Alignment," in *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, 2018.

[42] Liao, Kexin; Lebanoff, Logan; Liu, Fei, "Abstract Meaning Representation for Multi-Document Summarization," in *In Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, 2018.

[43] Bonial, Claire; Donatelli, Lucia; Abrams, Mitchell; Lukin, Stephanie M.; Tratz, Stephen; Marge, Matthew; Artstein, Ron; Traum, David; Voss, Clare, "Dialogue-AMR: Abstract Meaning Representation for Dialogue," in *In Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France, 2020.

[44] Elbasani, Ermal; Kim, Jeong-Dong, "AMR-CNN: Abstract Meaning Representation with Convolution Neural Network for Toxic Content Detection," *Journal of Web Engineering,* vol. 21, no. 3, pp. 677-692, 2022.