Towards information extraction from ISR reports for decision support using a two-stage learning-based approach

Dirk Mühlenberg[a*], Achim Kuwertz[a*], Philipp Schenkel[b], Wilmuth Müller[a]
(*equally contributed)
[a]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany;
[b]Karlsruhe Institute of Technology, Karlsruhe, Germany

ABSTRACT

The main challenge of computer linguistics is to represent the meaning of text in a computer model. Statistics based methods with manually created features have been used for more than 30 years with a divide and conquer approach to mark interesting features in free text. Around 2010, deep learning concepts found their way into the text-understanding research community. Deep learning is very attractive and easy to apply but needs massive pools of annotated and high quality data from every target domain, which is generally not available especially for the military domain. When changing the application domain one needs additional or new data to adopt the language models to the new domain. To overcome the everlasting "data problem" we chose a novel two-step approach by first using formal representations of the meaning and then applying a rule-based mapping to the target domain. As an intermediate language representation, we used abstract meaning representation (AMR) and trained a general base model. This base model was then trained with additional data from the intended domains (transfer learning) evaluating the quality of the parser with a stepwise approach in which we measured the parser performance against the amount of training data. This approach answered the question of how much data we need to get the required quality when changing an application domain. The mapping of the meaning representation to the target domain model gave us more control over specifics of the domain, which are not generally representable by a machine learning approach with self-learned feature vectors.

Keywords: text understanding, deep learning, transfer learning, AMR, data sparsity

1. INTRODUCTION

Timely and accurate intelligence is one of the key points for the commander's comprehensive situational awareness and, based on that, to gain a decent knowledge about his own options, on the one side, and the intents of his adversaries (red forces) for planning and conducting military operations, on the other side. Intelligence (gathering) is driven by the commander's information requirements and is commonly described by the Intelligence Cycle, a four-step cyclic process model [20]. The process starts with the commander issuing his intelligence requirements (during the *Planning* phase). These are then tailored to collection requirements to start the *Collection* phase, in which Intelligence, Surveillance & Reconnaissance (ISR) assets equipped with appropriate sensors will be tasked to collect the sensor data and information necessary to satisfy the intelligence requirements. Human interpreters then exploit the sensor data collected during ISR missions (in the context of single source processing). Results of these exploitation activities are then disseminated, e.g., in the form of military reports, as described by dedicated Standardization Agreements (STANAG, e.g., STANAG 3377 for Image Intelligence (IMINT)). The exploitation results are, from this point on, covert in textual form. Now, there is a certain effort needed to extract the relevant information from the texts in order to automate the following *Processing* step in the Intelligence Cycle. In this step, single-sourced information is collated, evaluated, analyzed, integrated and interpreted, getting higher-value multi-source products - which are then called (finished) intelligence. Computers can only support these manual steps if there is a way to extract the information contained in the textual reports as well as represent the information as objects and relations between them and store these resulting information graphs within a formal model (e.g., an ontology). This model then can be used as the technological base for building a common operational picture with some degree of computer support, mostly realized by assistance functionality. Military reports are mainly formulated as free text (like intelligence reports for Human Intelligence (HUMINT)). To extract the relevant information from such reports, we require natural language understanding techniques (NLU – a subfield of natural language processing (NLP), mostly dealing with semantic interpretation of text), which is a main task in the research field of computer linguistics.

Around 2010, deep learning techniques were introduced to the computer linguistics research community with great success. As all data driven techniques are mainly end-to-end techniques, deep learning needs massive amounts of high-quality data for training the models and to achieve good scores - as this approach is, in its pure mathematical sense, only regression analysis in a high-dimensional space, where many parameters have to be aligned to minimize some error function. This "data hunger" is an even greater obstacle in dynamic scenarios, where the domain model has to be adapted, e.g., when mission context changes, as it is the standard case for military operations in today's conflict areas.

To overcome the "data problem" for NLU tasks, we introduce a novel two-step approach, where we use a more generally trained linguistic base model (valid for the intended application area, i.e., defense & security) as an intermediate step and a partially engineered mapping for the more specific mission dependent application domains as a second step. As we use ontologies for the representation of the application domain, the mapping has the effect of populating the ontology with new instances as a result. For the semantic representation of linguistic information, we chose the approach of Abstract Meaning Representation (AMR). AMR represents semantics using a single rooted directed graph model, describing the meaning of a text in a very detailed fashion [1], where its nodes represent concepts (e.g., a person) and its edges depict roles (e.g., a location of sth.). The remainder of this paper is organized as follows. We start with related work in Section 2, describe AMR in Section 3, identify some shortcomings related to AMR in Section 4, introduce our two-step approach in Section 5, and finally conclude and discuss future work in Section 6.

2. RELATED WORK

The main task of our work is to extract the meaning of text and to map this information to a formal model (i.e., an ontology) to reason about. Our approach is to introduce a semantically rich linguistic model as an intermediate step (but not as an interlingua) to represent the extracted linguistic information in closed and general form, in contrast to more classical NLP approaches where the results of different chained NLP tasks are each represented on its own, e.g., as annotations in the analyzed document. The second step is to map the linguistic model to the intended target domain model. It will be shown that the choice of AMR is well aligned with the top-level domain model for intelligence analysis.

Related to our approach, there are several other approaches for information extraction from texts. One general direction here is for example to use a logical framework to formalize human language. One such line of research is about the attempt to formalize human language by means of a dedicated programming language for grammar definitions. This Grammatical Framework [4] allows defining multilingual parsers and generators by using a language independent representation of meaning.

Another idea is realized by the selection of only a subset of human language (e.g., a subset of English) with the requirement to reduce complexity and ambiguity. These subsets are called Controlled Natural Languages (CNL). There are versions of CNLs directly designed to serve as knowledge representation (such as the Attempto Controlled English [9], which is directly translatable to first order logic and, further, a subset of this language is even directly translatable to the Web Ontology Language (OWL)).

For our main task, human language in general is considered, thus the subset-based approaches (e.g., CNL) will not be sufficient for us. Also, manually defining the grammar for allowed textual information in form of a programming language is too restrictive in our case, since we cannot control the reports given as input. We need to address the problem the other way around, by learning to handle the linguistic properties common to the considered reports.

Our approach is based on transforming a given text into a formal representation of its meaning. To this end, different works exist that can be used as a basis (for the first step of our approach), as detailed in the following section.

3. AMR AS A LINGUISTIC MODEL FOR AN APPLICATION AREA

Candidates for the formal representation of general human language are divided into systems in the context of a pure linguistic level and systems working on a higher, ontological level.

The type of formal representation on a *linguistic* level uses syntax and semantics in the sense that is similar to the one used in computer science, where syntax defines the set of symbols (vocabulary) and semantics defines the principle of compositionality, expressed in valid combinations of the symbols of the vocabulary (in form of combination rules). Two representative candidates are given in the following:

- CCG (Combinatory Categorial Grammar) represents sentence logic by a grammar formalism, which is compositional and focused on constituents (the so-called categories or types like S for sentence or NP for noun phrase). The corresponding parsers are called *constituency* parser [8].

- UD (Universal Dependencies) represents sentence logic in a dependency structure in the form of dependency trees. The corresponding parsers are called *dependency* parsers (see [7] for an example).

Three representative candidates for the *ontological* level use the concept of meaning representation:

- UDS (Universal Decompositional Semantics) [5] is built upon UD by augmenting the syntactic dependency graphs with semantic annotations. The concept of semantic proto roles plays a central part for the correct identification of thematic roles (e.g., the distinction between roles such as AGENT and PATIENT), which goes back to a discussion of Dowty [6].
- In AMR (Abstract Meaning Representation) the representation of meaning is realized as a single rooted directed graph. AMR has a high semantic expressiveness (see [1]).
- UCCA (Universal Conceptual Cognitive Annotation) is representable as a directed acyclic graph (DAG), but uses a sparse semantic annotation model (only 12 categories) based on the concept of a scene, which is derived from the definition of a "clause" in basic linguistic theory [3].

The criteria for selecting a formal representation language of meaning (and respective parsers) in the context of our work were:

- semantic expressiveness of the intermediate linguistic model,
- the mere existence of parser software and
- the availability of high-quality training data.

Multilingualism (marked by the word "universal" in UD, UDS and UCCA) of the semantic parser models has no relevance in the context of our work, as the use of high-quality machine translation [10] is more predictable than coping with sparsely trained models in languages other than English or having no model for a certain language at all.

As CCG and UD are focused on the syntactic level, these approaches are also omitted from our research. For UCCA there exists, to the best of our knowledge, only one implementation called TUPA [11]. For AMR there exist more than four implementations at the time of writing, some of them are described and evaluated later in this paper. This rich code base gives us the possibility to select a proper parser in case of insufficient performance in a certain problem area.

The semantic model in the case of UCCA contains twelve categories. The intended goal is similar to the goal of AMR, i.e., to abstract away from certain syntactic constructions. But in direct comparison to AMR, UCCA falls short wrt. the expressiveness of the semantic model. Based on the abstract description of a scene, there are for example no means to express what is a named entity or what is a location in UCCA.

UDS is an interesting approach which is supported by the Decompositional Semantics (Decomp) Initiative. But at the time of writing, there is no parser available (only an announcement on the Decomp page under decomp.io) to use the UDS treebanks.

The evaluation of the approaches listed above by applying our selection criteria thus results in our choice of AMR.

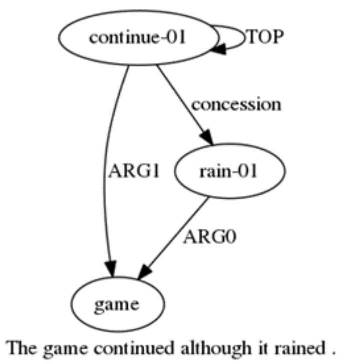

Figure 1: Graphical representation of AMR, created by means of an online-provided version of AMR Eager [12].

AMR used as an intermediate linguistic model

AMR uses a specification [1] where the meaning of text is captured in single rooted graphs. These graphs may be visualized in a graphical representation (see Figure 1 for an example) or merely in a textual style (see Figure 7 for an example). AMR makes extensive use of the Proposition Bank [2], where around 10000 verbal and nominal relations are listed together with their semantic core roles (these are the arguments of the verbal and nominal groups, each expressing a certain thematic role such as an actor or a theme). In addition to the Proposition Bank semantic core roles, AMR defines linguistic roles for modality, negation and many more. These are the so called non-core roles, which have direct relevance for real world aspects like named entities, locations, points or intervals of time, and which will play a central role for the mapping of the linguistic model to the domain model. A core feature of AMR is that it abstracts away from certain syntactic aspects (e.g., function words like prepositions or pronouns), as AMR is designed to capture the meaning of a sentence. More details regarding the AMR representation of meaning can be found in [2].

Evaluation measure SMATCH

The same research group which defined the AMR specification has also specified an evaluation measure which calculates the overlap between two semantic feature structures [13] (i.e., AMR graphs).

The SMATCH (short for semantic match) measure can be used to evaluate the performance of AMR parsers. This measure is used to compare the similarity of two AMR graphs on a structural level. As a result, a value in the interval [0, 1] is returned, where a value of 1 indicates structural correspondence for two AMR graphs. The performance of a parser can be evaluated using SMATCH by providing the sentence for a test datum as input to the parser and comparing the AMR graph generated as output with the expected AMR graph of the test datum. The overall performance of the parser can then be calculated by averaging the SMATCH values over all data of the test dataset.

For two given AMR graphs G_1 and G_2, the SMATCH value s is calculated as

$$s(G_1, G_2) \coloneqq max_{\phi} F_1^{\phi}(G_1, G_2), \tag{1}$$

where ϕ represents a mapping $\phi : V_1 \rightarrow (V_2 \cup \emptyset)$ of the node set V_1 of the graph G_1 to the node set V_2 of the graph G_2 (or the empty set). This mapping determines which nodes and edges are compared when evaluating the similarity of the AMR graphs. To determine the SMATCH value, the maximum F_1 [15] value under all possible mappings ϕ of the nodes has to be computed. The F_1 value describes the structural similarity of the AMR graphs. For a given mapping ϕ, the number of matching nodes and edges in the two graphs is determined, and then used as accuracy according to the total number of nodes and edges in G_1 or as a hit rate according to G_2. The F_1 value then results as the harmonic mean of the accuracy and hit rate values. In Figure 2 the SMATCH evaluates to 0.75.

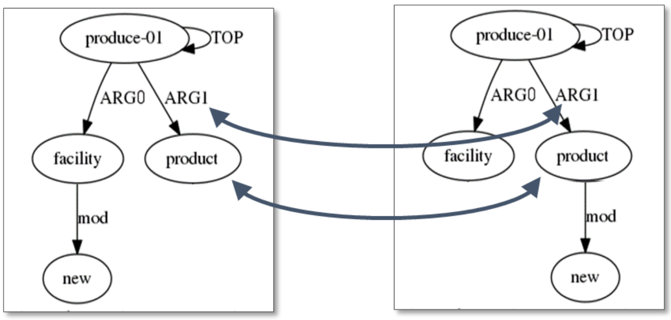

Figure 2: SMATCH compares concept nodes and edge labels using the F_1 score for a given mapping, and returns the maximized score over all mappings.

Parser concepts and implementations

There are different parser implementations available for AMR, which are based on different concepts and approaches of translating text into AMR graphs. The next paragraphs describe four parsers evaluated in the context of this work.

JAMR was the first parser implementation available [14]. JAMR uses a two-phase approach for concept and relation identification. The concept identification, i.e. mapping terms in a sentence to AMR nodes as well as to Proposition Bank entries, in the first phase is done by optimizing a feature function that connects word spans (partitions of the sentence) with possible AMR fragments known from the training phase. The second phase is the relation identification, where the correct connections of nodes and their labels (in term of AMR roles, both core and non-core roles) are retrieved. This phase builds upon the AMR fragments from the first phase by searching the maximum spanning connected subgraph which is a derivation of the standard maximum spanning tree algorithm.

CAMR [16] also uses a two-phase approach, where a linguistic tree structure (parse tree) is extracted by a dependency parser and, based on this, a transition system is defined which applies a finite set of transformations to map the dependency tree to the appropriate AMR graph. Both phases are independent of each other and may be trained separately. The approach is based on the observation that both representations (parse trees and AMR graphs) are structurally very similar and a transition can be accomplishable with a finite set of transformation steps. The training for the transition phase can be done on a more limited set of tree-graph pairs (wrt. their number) as training data compared to the set of training data needed to train the dependency parser itself.

AMR Eager [17] is a transition based parser inspired by the dependency transition system ARCEAGER of Nivre [18]. Like CAMR the approach is based on the observation that dependency trees and AMR graph are similar in structure.

Neural AMR [19] is a sequence-to-sequence encoder-decoder neural network architecture, which is trained end-to-end with careful data preprocessing (anonymization of rare words like dates or named entities) to fight the out-of-vocabulary (OOV) problem as training data is sparse. A second technique is to bootstrap a high-quality AMR parser by incorporating unannotated sentences from the Gigaword corpus (available from Linguistic Data Consortium (LDC) of the University of Pennsylvania) for self-training.

The aforementioned parsers are altogether trained on a data set available from the LDC and, hence, their performance is directly comparable. This LDC set contains about 20,000 sentence-to-AMR pairs.

Parser performance analysis

The comparison of the mean SMATCH value (see Figure 3 for our own evaluation results, [17] & [19] for reference) shows a similar performance for the four parsers JAMR (version of 2016), CAMR, AMR Eager and Neural AMR, with Neural AMR being only ten percent behind the top performer JAMR. This is a surprisingly good result for an end-to-end trained neural network solution on such a sparse data set. The two methods used to fight data sparsity in Neural AMR allow a quality gain of nearly ten percent compared to the version without any tuning measures.

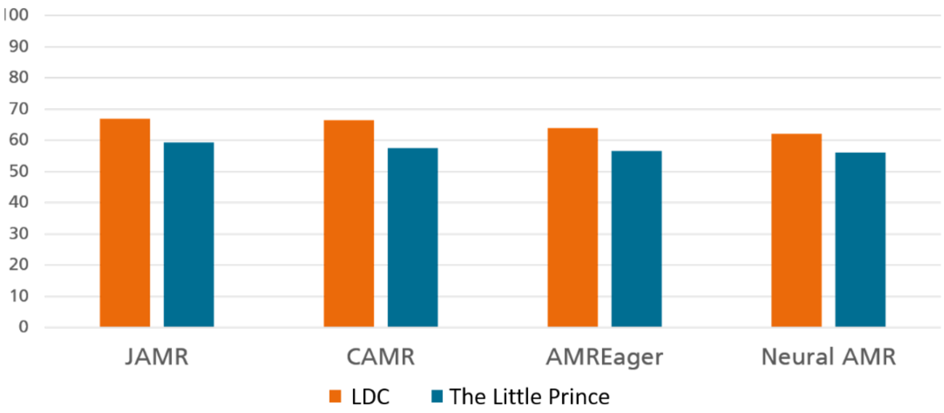

Figure 3: Mean SMATCH values (as percentages) for the four considered AMR parsers, evaluated on two different data sets containing AMR-annotated sentences, namely (a part of) the LDC data set and the Little Prince data set. (The parsers were trained on parts of the LDC data set).

As Figure 4 clearly indicates, despite the similar mean SMATCH scores, the four parsers differ considerably when analyzing detailed SMATCH values related to the recognition of only certain AMR non-core roles in given texts. The four parsers here are tested on another set (based on "The Little Prince" data set) containing only sentences in which either the tested parser recognized the targeted role or the reference AMR graph contained the targeted role. As expected, Neural AMR performs poorly for named entities, as the performance of this parser is highly sensitive to the occurrence of unseen words (OOV condition), but the parser performs well for the other considered cases (especially, for the non-core role *condition*). On the other side, CAMR is very bad at recognizing negations in sentences, which might be a real show stopper for using this parser for information extraction tasks.

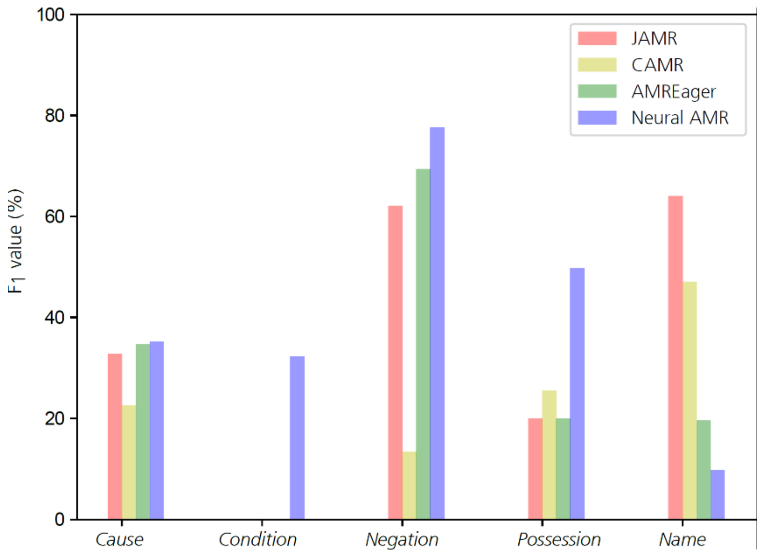

Figure 4: Comparison of the four considered AMR parsers regarding their performance in recognizing certain semantic parts of given sentences (mapping to the five depicted AMR non-core roles). Performance is measured as the mean F_1 score (given as percentages) regarding the detection of roles (i.e., sentence parts) by the parsers over the Little Prince data set.

This short performance evaluation shows that the choice of a certain parser is a real trade-off between the performances for the recognition of different AMR roles. In summary, one can state that the quality for a comprehensive information extraction is yet too low to use this technology in the context of critical application areas, and this is only partially based on the fact that training data is rare. As the manual creation of qualitative AMR graphs as training data is expensive (especially due to some flaws in the specification which can lead to ambiguities in the model), we examined the dependency of the quality (measured by SMATCH) from the number of training data pairs, to get an overall guidance regarding how many AMR graph need to be created as training data for satisfying a certain quality requirement.

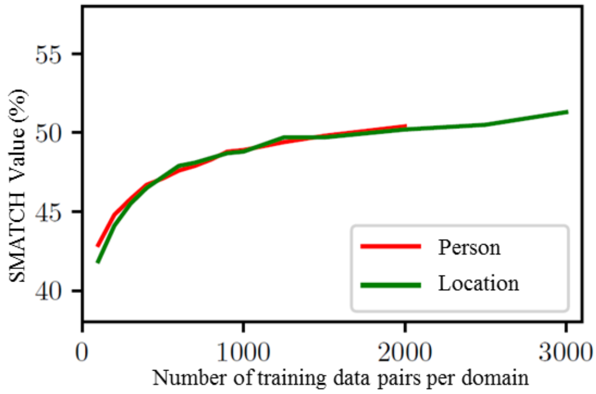

Figure 5: Dependency of the SMATCH value for training an AMR parser on the number of sentence-AMR pairs used as training data. Depicted is the mean SMATCH value over the LDC data set, as a percentage.

Figure 5 shows a diagram where the mean SMATCH value is plotted against the number of training data pairs used. The diagram is recorded for the AMR Eager parser. This is done for a partition of the LDC data set, where one part contains *Person* concepts and the other part contains *Location* concepts exclusively. It can be clearly seen that for both domains the functional dependency is nearly the same and that the slope ("increase in SMATCH value per training pair") is drastically decreasing for numbers of pairs greater than 1000. Beyond that point, the slope is nearly constant. Yet, an extension beyond 3000 was not possible as there were no more data available from LDC data set for this setting considering two domains.

Military Intelligence Analysis

This work is done in the context of military intelligence. A well-known top-level model to describe the domain relevant for the intelligence analysis tasks of an intelligence officer is the Military Intelligence Pentagram [20]. It introduces five top-level knowledge categories, depicted as the corners of a pentagram (Event, Bios etc.), and additionally a concept for time, depicted in between the corners. All corners are fully connected. This layer describes the topmost level of our intelligence management ontology (i.e., our target domain model).

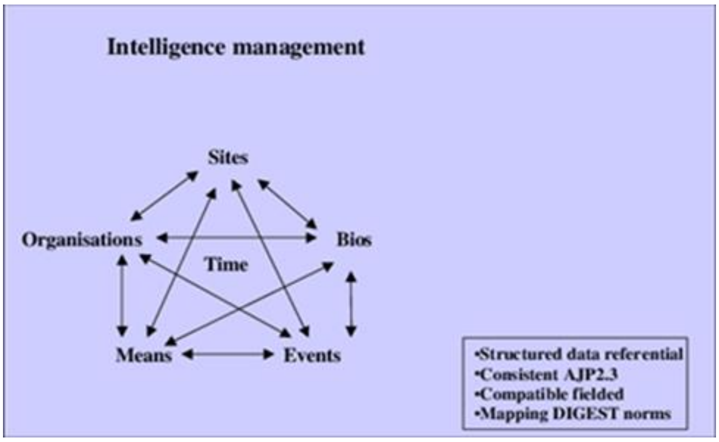

Figure 6: Intelligence Pentagram [20].

The collation step in the *Processing* phase of the Intelligence Cycle is responsible for grouping together related items of information or intelligence to provide a record of *events*, to facilitate further processing. So the event concept is central for the processing phase (although not emphasized in the Intelligence Pentagram). In the analysis step, isolated events may be chained up, e.g., in a timeline or correlated into different dependency relations like causal dependency.

In the context of this work we use the formal concept of ontologies. An ontology is a formal and explicit specification of a shared conceptualization [21]. There are already some works in the area of military applications based on ontologies as knowledge models, a short overview is given in [22]. Herein, their ONTO-CIF model uses the Intelligence Pentagram as the central hub (top-level) of their intelligence ontology.

4. SHORTCOMINGS RELATED TO AMR

As part of our performance evaluation for AMR parser, a shortcoming of the SMATCH evaluation measure has come to light. Rating the similarity of AMR graphs (as representations of sentence meaning) based on the SMATCH measure is, at least in some parts, contra-intuitive regarding the semantic similarity of two sentences. The AMR specification allows modeling the same sentence meaning with differently structured AMR graphs, e.g., using the concept of reification (substitute a role by a concept) or using inverse relations. This ambiguity poses difficulties for a correct evaluation of parser performance wrt. to the comparison of resulting AMR graph. The evaluation measure SMATCH in its current version is not yet taking this language features into account. This can result in the AMR graphs of two sentences, which would intuitively be considered as semantically similar or even equal, to be rated as being different – as illustrated in Table 1.

Table 1: The sentence "The girl left because the boy arrived." in AMR without and with reification (from [1]).

(l / leave-11 :ARG0 (g / girl) :cause (a / arrive-01 :ARG1 (b / boy)))	(l / leave-11 :ARG0 (g / girl) :ARG1-of (c / cause-01 :ARG0 (a / arrive-01 :ARG1 (b / boy))))

Comparing the two AMR graphs representing the sentence "The girl left because the boy arrived." in Table 1 evaluates to a SMATCH value of 0.78, albeit they represent the same meaning. The use of reification also increases the effort to map from the intermediate linguistic model to the target domain model as the mapping rules have to respect all possible (allowed) representations of a sentence meaning.

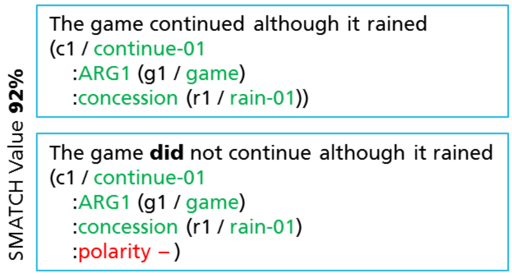

Figure 7: Textual AMR examples showing the influence of negation (polarity) on SMATCH values.

The principle of SMATCH as a semantic similarity measure yet has another basic shortcoming from an application perspective, as indicated in the example shown in Figure 7. The second sentence in Figure 7 is a negation of the first sentence, so one would expect that the similarity value should be very low (from an intuitive semantic point of view). But SMATCH evaluates to 92%, as the two AMRs only differ in one node (the polarity node). This fact becomes even more problematic in combination with the first shortcoming, since e.g. the AMR graphs for semantically similar sentences can have lower SMATCH values than those for sentences with negated meaning.

5. MAPPING OF AMR TO DOMAIN MODEL

The automatic transformation from text to a linguistic model via an AMR parser allows the structural representation of the meaning of the text with semantic features like semantic core roles (the arguments of a Proposition Bank entry) and non-core roles (such as *:time* or *:location* role) in AMR. Our domain model is based on the six categories from the Intelligence Pentagram. Table 2 shows a possible mapping between the aforementioned intelligence categories (first row) and some of the non-core roles of AMR (second row).

Pentagram	Bios	Sites	Organisations	Means	Events	Time
AMR (Concept, Role)	Named Entity	:location	Named Entity	:means	Prop-Bank	:time, :duration

Table 2: Mapping of AMR concepts and roles to concepts in the Intelligence Pentagram (partial).

This is the conceptual starting point of our mapping from an AMR graph, as created by a parser, to a part of our intelligence ontology. This mapping defines functions to transform the information contained in the AMR graph into instances (not necessarily new ones, see the data association problem) in the knowledge base of the ontology. At a first glance it seems to be a relatively simple task, but a generally functioning mapping is hard to realize, which is detailed in the following. For this, we divide this general mapping task into several sub-tasks, which we are handling sequentially. First of all this task is generally speaking a functional mapping from one graph (AMR) to another graph (part of the knowledge base represented with a graph structure). In this context we identified the following separate challenges, which we detail here as part of our mapping concept:

- Identification of *Event* by analyzing the selected Proposition Bank entry and mapping this to an existing *Event* category in the target model. As the Proposition Bank contains around 10000 entries, there is the need for a classification of the detected verb group in terms of the existing event categories in the target model. This can be done by means of using the SUMO ontology [30] in combination with WordNet [31]. SUMO has a sub-model for e.g. military processes. If the verb or one of its derivations (synonym or noun form derived from WordNet) is found in SUMO, this is a candidate for an event category [22].

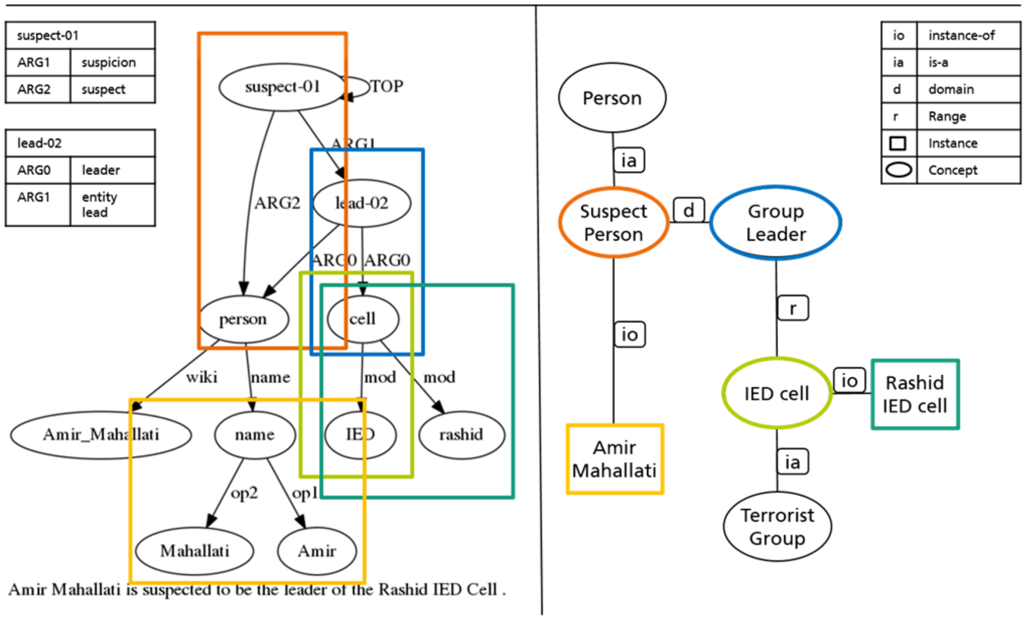

Figure 8: Example of frame association from AMR graph to ontology graph. Colored rectangles on the left indicate the mapping of frames to the appropriate ontology concepts and instances on the right.

- Frame association or mapping discovery, which requires to find the regions in both knowledge graphs which are candidates for an optimal semantic match. This is only needed if the AMR role or concept to be mapped is complex like in the example above (see Figure 8), where the sentence “Amir Mahallati is suspected the leader of the Rashid

IED cell” is parsed into AMR and afterwards mapped to the target model. This example is of simple structure but already shows some of the challenges that have to be addressed:

- Generally, it is hard to decide what a concept is and what is an instance (except in the case of named entities).
- For (suspect-01 *:ARG2* person), a simple mapping to the *SuspectPerson* concept can be performed by using a so called derivative function from WordNet as mentioned above to match via a simple string comparison.
- For (lead-02 *:ARG0* cell), there is a similarity with *GroupLeader*, here the similarity between *cell* and *group* can e.g. be derived using word2vec based vector space models (see paragraph about semantic similarity).
- (cell *:mod* IED) is a simple string mapping to the “IED cell” concept although the sequence of words is reversed,
- but (cell *:mod* IED *:mod* rashid) is harder to recognize as an instance “Rashid IED cell” of the concept “IED cell”. It is only identifiable as an instance if *Rashid* can be recognized as the name of a city.
- In the context of this work, we start with a fixed set of engineered functions to implement the mapping of frames from AMR to our ontology. This approach may be further automated in the future by using Bayesian decision theory with incorporating a “meaning” feature set containing names of concepts and relations, manual annotations (comments) as well as structural aspects (e.g., an *Address* commonly contains city, street, zipcode and so on) and the axiomatic background of each model [24].

- Semantic similarity – in the previous stage the regions in the two models which are conceptually similar are identified. The following steps have to manage the transformation of knowledge from the AMR representation into the knowledge base. For this we first focus on the structural aspect of word groups and then on the level of single concepts (words).
    - Information management on the base of topological similarity – based on the region mapping, the parts of the AMR graph describing entities can be compared to the conceptual parts of the knowledge base, e.g. mapping a (AMR) person with a name to a new instance of *Person* in the knowledge base.
    - Word similarity – there are several publications for semantic similarity on a conceptual level, which are based e.g. on statistical co-occurrence of words in (web) documents (see [26]) or based on taxonomies, where similarity is measured by the information content of the least common subsumer of two concepts in a taxonomy (see [25]). With the work about the concept of word embedding (implementations introduced in 2013, see word2vec [27] and GloVe [28]), there is a very powerful - unsupervised and online trainable - statistical linguistic vector space model, which is capable of computing similarities on the base of the *context* of a word (generally the context of the word is the sentence which contains it). So two words which are used in the same context are similar. This model is superior to simple Tf-idf (term frequency – inverse document frequency) vector space model or to mainly manually engineered taxonomies e.g. contained in ontologies, as it is very easy to train and the linguistic model is able to learn the context of a word. In the context of this work, we intend to apply word embeddings to fulfill the task of similarity computation on concept (word) level.
- Data format translation is a sub-task which manages the format translations for all data given in different representations, such as time and geolocation or other metric data types. Of special interest are enumerated types – nominal and ordinal – which may have different representations in the two models.
- The information integration problem is always apparent, when there is the need to insert a new instance or update an existing one in the knowledge base. To perform this operation one has to decide if this instance is a new one and should be associated (data association) correctly with its context in the model or if it already exists somewhere in the model and some values (attributes) of the instance have to be updated. In the latter case, we have an information fusion problem, where certain values are updated according to actual observations [29]. Both problems can be tackled using a semantic world model [29] for managing the information in the knowledge base.

Implementing the aforementioned concept and principles for the second transformation step of our proposed approach in full is part of our future work.

6. DISCUSSION AND FUTURE WORK

In this paper, we presented our work performed wrt. to information extraction from textual reports (in the military domain using a newly proposed two-step approach for this extraction based on an intermediate semantic representation (AMR in our case). We evaluated the application of existing state-of-the-art AMR parser (as part of the first step) for use in the intended application domain, in general and in more detail (regarding the recognition of important semantic sentence parts),

and reported on several shortcomings of applying the SMATCH evaluation measure (in combination with the AMR specification) to our application domain. In addition, we examined the relation of training set size to AMR parser performance in order to be able to determine to required scale of training samples for reaching a targeted performance.
Further work concerns the full implementation of the outlined mapping from AMR graphs to our domain ontology, as well as commencing to address the mentioned semantic shortcomings of measures for comparing AMR graphs wrt. to our application domain. This can range from normalizing considered AMR graphs, prior to comparing them, over more semantically aware evaluation measures to defining a stricter set of rules for creating AMR graphs (especially when creating data sets specific for our application domain).

REFERENCES

[1] Banarescu L. et al., Abstract Meaning Representation (AMR) 1.2.5 Specification, 23. Feb. 2018, https://github.com/amrisi/amr-guidelines/blob/master/amr.md (19.02.2019).

[2] Palmer M., Gildea D., and Kingsbury P., “The Proposition Bank: A Corpus Annotated with Semantic Roles”, Computational Linguistics Journal, 31:1 (2005).

[3] Abend O. and Rappoport A., “Universal Conceptual Cognitive Annotation (UCCA)”, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 228-238 (2013).

[4] Ranta A., “Grammatical Framework: Programming with Multilingual Grammars”, Center for the Study of Language and Information/SRI, Stanford (2011).

[5] White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R.,, Rawlins K, and Van Durme, B., “Universal Decompositional Semantics on Universal Dependencies.”, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1713–1723 (2016).

[6] Dowty D., “Thematic proto-roles and argument selection.” Language, 67(3), 547–619 (1991).

[7] Kiperwasser E., and Goldberg Y., “Simple and Accurate Dependency ParsingUsing Bidirectional LSTM Feature Representations”, Association for Computational Linguistics (2016).

[8] Steedman, M. and Baldridge, J., “Combinatory Categorial Grammar.“, in R. D. Borsley & K. Börjars (Ed.), Non-Transformational Syntax: Formal and Explicit Models of Grammar, 181–224 (2011).

[9] Fuchs N.E., Kaljurand K., and Kuhn T., “Attempto Controlled English for Knowledge Representation”, Reasoning Web 2008, LNCS 5224, 104–124 (2008).

[10] Hassan H. et al., “Achieving Human Parity on Automatic Chinese to English News Translation”, arXiv:1803.05567 (2018).

[11] Hershcovich D., “Transition-based UCCA Parser”, 30.06.2018, https://github.com/danielhers/tupa (2019).

[12] Damonte M. and Cohen S.B., “Abstract Meaning Representation parser”, the cohort, http://cohort.inf.ed.ac.uk/amreager.html (2019).

[13] Cai S. and Knight K., “Smatch: an Evaluation Metric for Semantic Feature Structures”, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Vol. 2, 748-752 (2013).

[14] Flanigan J. et al., “A Discriminative Graph-Based Parser for the Abstract Meaning Representation”, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, 1426–1436, Baltimore, Maryland (2014).

[15] Sørensen, T., "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons". Kongelige Danske Videnskabernes Selskab. 5 (4): 1–34 (1948).

[16] Wang C., Xue N., and Pradhan S., “A transition-based algorithm for AMR parsing”, Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 366–375, Denver, Colorado, May–June, Association for Computational Linguistics (2015).

[17] Damonte M., Cohen S. B., and Satta G., “An incremental parser for abstract meaning representation”, CoRR, vol. abs/1608.06111, (2016).

[18] Nivre J., “Incrementality in deterministic dependency parsing”, Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together, ACL (2004).

[19] Konstas I., Iyer S., Yatskar M., Choi Y., and Zettlemoyer L., “Neural AMR: sequence-to-sequence models for parsing and generation”, CoRR, vol. abs/1704.08381 (2017).

[20] Biermann, J. et al., “From Unstructured to Structured Information in Military Intelligence: Some Steps to Improve Information Fusion” (2004).

[21] Gruber, T., “A translation approach to portable ontology specifications”, Knowledge Acquisition, 199-220, (1993).
[22] Dragos V., “Developing a core ontology to improve military intelligence analysis”, International Journal of Knowledge-Based and Intelligent Engineering Systems, IOS Press, 2013, 17 (1), 29-36 (2013).
[23] Chow C. and Jonathan I&W, “Integration of Linguistic Resources for Verb Classification: FrameNet Frame, WordNet Verb and Suggested Upper Merged Ontology”. 4394. 1-11. 10.1007/978-3-540-70939-8_1 (2007).
[24] Tang J., Li J., Liang B., Huang X., Li Y., and Wang K., “Using Bayesian Decision for Ontology Mapping”, Journal of Web Semantics, Volume 4, Issue 4, 243-262 (2006).
[25] Sánchez D., Batet M., Valls A., and Gibert K., “Ontology-driven web-based semantic similarity”, Springer Science + Business Media, LLC (2009).
[26] Bollegala D., Matsuo Y., and Ishizuka M., “WebSim: A Web-based Semantic Similarity Measure”, The 21st Annual Conference of the Japanese Society for Artificial Intelligence (2007).
[27] Mikolov T., Corrado G., Chen K., and Dean J., “Efficient Estimation of Word Representations in Vector Space,” Proceedings of the International Conference on Learning Representations (ICLR 2013), pp. 1–12 (2013).
[28] Pennington J., Socher R., and Manning C.D., “GloVe: Global Vectors for Word Representation”, Empirical Methods in Natural Language Processing (EMNLP), 1532—1543 (2014).
[29] Kuwertz A., Mühlenberg D., Sander J., and Müller W., “Applying Knowledge-Based Reasoning for Information Fusion in Intelligence, Surveillance, and Reconnaissance”, in Sukhan L., Hanseok K., and Songhwai O. (Eds.), Multisensor Fusion and Integration in the Wake of Big Data, Deep Learning and Cyber Physical Systems, 119-139, Springer (2017).
[30] Niles, I., and Pease, A., “Towards a Standard Upper Ontology.”, Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds, Ogunquit, Maine (2001).
[31] Miller G.A., “WordNet, A Lexical Database for English”, https://wordnet.princeton.edu (2019).