Abstract Meaning Representation (AMR) is a relatively new addition to the latent space of meaning representations. It divorces itself, by design, from the previous collective multi-layer representations of a variety of linguistic phenomena—primarily from the syntactic representation which has been so central to NLP over the past two decades. And claims to be a complete representation of a sentence. That said, it does build upon some long-standing, core linguistic phenomena, and adds a few layers of abstraction. It was originally designed with an aim to improve machine translation. We will look at a few AMRs and how it differs from what existed before it was proposed and briefly discuss its merits and shortcomings. We will also look at the evaluation metric Smatch which stands for semantic match.

Next, we will present our work on an AMR parser---CAMR. This is an open-source, transition based dependency tree to graph transducer which has been a collaborative effort between Brandeis University and cemantix.org. It was the best performing system at the SemEval-2016 shared task on meaning representation parsing and tied with another system that was an ensemble which included CAMR. Three other participating systems at SemEval-2016 built upon CAMR.

CAMR represents a two-stage framework that takes the sentence as input and produces its AMR. In its first stage, the algorithm parses the sentence using a dependency parser. In the second stage, it learns a sequence of transitions while walking the dependency tree and transfers it into an AMR graph. It is a supervised algorithm which during training performs an optimal alignment between the tokens in the sentence and the concepts in the corresponding AMR to produce a span graph. A set of transitions are trained to convert the dependency tree into this span graph. During testing it produces this span graph and then deterministically transforms it into the final AMR. There are several advantages using this approach. First, the dependency parser can be trained on a training set much larger than the training set for the tree-to-graph algorithm, resulting in a more accurate AMR parser over-all. This is evidenced from the first version of CAMR which yielded a 5% absolute improvement in Smatch F-score over the previous best result. Second, the actions that we design are linguistically intuitive and capture the regularities in the mapping between the dependency structure and the AMR of a sentence. Third, it runs in nearly linear time in practice in spite of a worst-case complexity of $O(n^2)$.

We also report recent enhancements to the parser which included a new infer action to address abstract concepts and the addition of richer, complementary, features generated by models trained on the various layers of the much larger OntoNotes corpus. These include semantic roles, coreference, rich named entities, and word senses. The resulting system's AMR parsing performance was 7% absolute in Smatch F-score over the initial version of CAMR.

The parser code is available at: https://github.com/c-amr/