Approaching the Source of Symbol Grounding with Confluent Reductions of Abstract Meaning Representation Directed Graphs

Anonymous ACL submission

Abstract

Abstract meaning representation (AMR) is a semantic formalism used to represent the meaning of sentences as directed acyclic graphs. In this paper, we describe how real digital dictionaries can be embedded into AMR directed graphs (digraphs), using state-of-the-art pretrained large language models. Then, we *reduce* those digraphs in a *confluent* manner, i.e. with transformations that preserve their circuit space. Finally, the properties of these reduced digraphs are analyzed and discussed in relation to the symbol grounding problem.

1 Introduction

When we encounter a word with unknown meaning, we look up its definition. If, in the definition, we find another unkown word, then we can look it up too, repeating this process until it eventually stops. However, a language cannot be fully learned from a dictionary via definition look-up alone: *The meaning of some words has to be acquired beforehand*. Explaining how and why humans have the unique capacity to *ground in arbitrary symbols the things in the world they refer to* in order to break this circularity is known as the symbol grounding problem (Harnad, 1990, 2024). One might then ask : how many words – and which words – are sufficient to define the rest? Given a dictionary, a set of words of that dictionary that is sufficient to define all remaining words is called a *grounding set* and has been shown to be formally equivalent to a feedback vertex set (FVS) when representing dictionaries as directed graphs (Blondin Massé et al., 2008). Furthermore, it turns out that grounding sets of minimal cardinality are far from being unique, although they all present consistent psycholinguistic characteristics, such as being more frequently used, learned at a younger age, and more abstract (Vincent-Lamarre et al., 2016).

A major issue when representing dictionaries as graphs is *word sense disambiguation* (WSD): When a word used in a definition itself has more than one definition, how can we *algorithmically* identify which is the intended one? To address this, Vincent-Lamarre et al. assumed the first meaning to be used whenever the word appeared in a definition (Vincent-Lamarre et al., 2016). Since the advent of transformer architecture in (Vaswani, 2017) however, new methods for addressing the challenges have become available. This work investigates whether embedding dictionary definitions into Abstract Meaning Representations (AMR) (Banarescu et al., 2013) can help resolve some of the limitations of previous methodologies. As AMR draws from Propbank Frames when possible, generated embeddings might provide unique labels to different definitions of a given word, partially solving the issue of WSD. Additionally, AMR can be examined as a more atomic representation of meaning. Minimal grounding sets of dictionaries embedded in AMRs could provide insights into a broader question of *what concepts need to be known in order to learn all others* but also *what relationships between these concepts have to be understood beforehand*. In addition to using the AMR formalism, we study in more depth the effect of applying MFVS-preserving reductions (Levy and Low, 1988; Lin and Jou, 2000; Lemaic, 2008; Kiesel and Schidler, 2022) on directed graphs from a psycholinguistical point of view.

The paper is organized as follows. Section 2 introduces the necessary definitions and notation. Section 3 describes our methodology for embedding definitions in AMR digraphs. Section 4 presents the digraph reductions. The experiments and their results are described in Section 5. We then discuss in Section 6 the results before concluding.

2 Preliminaries

This section introduces the necessary definitions and notation for the following sections. In Subsec-

tion 2.1, basic graph theoretical concepts are briefly recalled. Subsection 2.2 explains how dictionaries can be formally represented and transformed into directed graphs. Finally, Subsection 2.3 summarizes the main ideas of abstract meaning representation (AMR).

2.1 Directed Graphs

A *directed graph* or *digraph* is an ordered pair $G = (V, A)$, where V is a set whose elements are called *vertices* and $E \subseteq A \times A$ is a set whose elements are called *arcs*. Another digraph $H = (V', A')$ is called a *subgraph* of G if $V' \subseteq V$ and $A' \subseteq \{(u, v) \in A \mid a, u \in V'\}$. Let $G = (V, A)$, $u, v \in V$ and $U \subseteq V$. We say that u is a *predecessor* (resp. *successor*, resp. *neighbor*) of v if $(u, v) \in A$ (resp. $(v, u) \in A$, resp. $(u, v) \in A$ or $(v, u) \in A$). The set of predecessors (resp. successors, resp. neighbors) of u in G is denoted by $N_G^-(u)$ (resp. $N_G^+(u)$, resp. $N_G(u)$). Similarly, we say that the arc a is an *incoming* (resp. *outgoing*, resp. *incident*) arc of u if $a = (u, v)$ (resp. $a = (v, u)$, resp. $a \in \{(u, v), (v, u)\}$) for some vertex v. The set of incoming (resp. outgoing, resp. incident) arcs of u in G is denoted by $A_G^-(u)$ (resp. $A_G^+(u)$, resp. $A_G(u)$). The set U is called a *directed clique* or *diclique* of G if $(u, v) \in A$ for all $u, v \in U$, $u \neq v$, and $(u, u) \notin U$, for all $u \in U$. An arc $(u, v) \in A$ of G is called *bidirectional* if $(v, u) \in A$. The set of all bidirectional arcs of G is denoted by $A^{\leftrightarrow}$.

Given two vertices $u, v \in U$, a *uv-path of G of length k* is a sequence $p = (u_1, u_2, \ldots, u_{k-1})$, where $u = u_1$, $v = u_k$ and $(u_i, u_{i+1}) \in A$ for $i = 1, 2, \ldots, k-2$. We say that p *visits* a vertex x (resp. an arc (x, y)) if $x = u_i$ for some $i \in \{1, 2, \ldots, k\}$ (resp. $(x, y) = (u_i, u_{i+1})$ for some $i \in \{1, 2, \ldots, k-2\}$). If $u = v$, then p is called a *circuit*. An arc is called *acyclic* if there does not exist any circuit of G that visits it. Similarly, G is called a *directed acyclic graph* (DAG) if all its arcs are acyclic. A digraph is called *strongly connected* if for all $u, v \in V$ there exist at least one uv-path and at least one vu-path in G. A *strongly connected component* of a directed graph $G = (V, A)$ is a subgraph $H = (V', A')$ of G that is strongly connected and such that V' and A' are maximal, meaning that no additional vertex or arc of G can be included in H without breaking its property of being strongly connected.

It is often convenient to consider digraphs obtained by removing some of its vertices or some of its arcs. Let $G = (V, A)$ be a digraph, $u, v \in V$. Then $G - u = (V', A')$ where $V' = V - \{u\}$ and $A' = A \cap V' \times V'$. Similarly, $G - (u, v) = (V, A - \{(u, v)\})$. The notation is extended to sets of vertices and sets of arcs: If $U \subseteq V$ and $B \subseteq A$, then $G - U = (V', A')$, where $V' = V - U$ and $A' = A \cap V' \times V'$ and $G - B = (V, A - B)$. Finally, $G \circ u = (V', A')$, where $V' = V - u$ and $A' = (A - A_G(u)) \cup (N_G^-(u) \times N_G^+(u))$.

If $G = (V, A)$ is a digraph and U a subset of V, we say that U is a *feedback vertex set* (*FVS*) of G if, for every circuit $c = (v_1, v_2, \ldots, v_k, v_1)$, there exists an index i such that $v_i \in U$. Equivalently, U is a feedback vertex set of G if $G - U$ is acyclic. A *minimum feedback vertex set* (*MFVS*) of G is a feedback vertex set of minimum cardinality. The problem of identifying MFVS in directed graphs is NP-hard and has been thoroughly studied in the last 40 years (Karp, 2010).

2.2 Dictionaries

In its most simple form, a dictionary is simply a set of lexical units with a map associating to each lexical unit a definition. Some dictionaries have stronger structure than others. In that spirit, we distinguish two types of dictionaries in this paper, as described by Definitions 1 and 2. Let $\mathcal{S}$ be the set of all strings on a given alphabet.

Definition 1. *A* raw *dictionary is an ordered pair $D = (L, d)$, where L is a finite set of* lexemes *and $d : L \to S$ is a map associating each lexeme to a nonempty string called its* raw definition.

Although raw dictionaries are useful to human, their structure is quite limited when one wishes to study the definitional relations between lexemes. A richer representation can be used. Given a set A, let $\mathrm{Seq}(A)$ be the set of all finite sequences of elements taken in A.

Definition 2. *A* disambiguated *dictionary is an ordered pair $D = (L, d)$, where L is a finite set of* lexemes *and $d : L \to \mathrm{Seq}(L)$ is a map associating each lexeme to a nonempty sequence of lexemes called its* disambiguated definition.

Disambiguated dictionaries can conveniently be represented by a digraph: It suffices to represent each lexeme in L by a vertex, and add an arc between two lexemes ℓ_1 and ℓ_2 whenever $\ell_1 \in d(\ell_2)$, i.e. ℓ_1 is a lexeme occurring in the definition of ℓ_2. Unfortunately, to the best of our knowledge, disambiguated dictionaries are not available in practice and must be built using natural language processing

tools. As we show in Section 3, abstract meaning representation seems to be well adapted to this task.

2.3 Abstract Meaning Representation

Abstract meaning representation (AMR) is a semantic formalism designed to represent the *meaning* of a given sentence, whilst abstracting away from syntactic idiosyncrasies (Banarescu et al., 2013). In this formalism, sentences are represented by rooted, directed acyclic graphs where vertices represent *concepts* and labelled arcs are used to represent *semantic relationships* between them. Those concepts are often lifted from Propbank (Palmer et al., 2005), a dataset with disambiguated concepts and arguments, which are specific relationships a given concept can have with other concepts. An interesting property of AMR is that different sentences with the same meaning should have the same AMR. For example, the sentences "apple is defined as a red round fruit" and "a red round fruit is the definition of apple" share the AMR representation shown in Figure 1. AMR has been of help for a variety of use-cases such as text summarization (Dohare et al., 2017; Kouris et al., 2024), machine translation (Li and Flanigan, 2022) and grounded human-robot conversations(Bonial et al., 2023). For a more complete review of AMR applications, refer to (Tohidi and Dadkhah, 2022).

This semantic formalism is of interest for embedding definitions for two main reasons. First is the potential for word-sense disambiguation. In Propbank, words with multiple meanings are assigned numbered senses to distinguish between each meaning. For example, the word *admonish* has two distinct frames, each associated with specific arguments. On one hand, admonish-01, whose meaning is *persuade warningly*, takes 3 arguments: the persuader (ARG0), the persuaded agent (ARG1) and the persuaded action (ARG2). On the other hand, admonish-02, whose meaning is *chastise* takes 3 arguments as well, but with different meaning: the chastiser (ARG0), the chastised (ARG1) and the wrongdoing (ARG2).

In the previous example, disambiguation could be accomplished in two manners. The first one is via the numbered labels : when facing multiple definitions for a given word, assign when possible a unique numbered label to represent specific meaning. The other option is by translating it to another defined concept entirely. AMR is like translating from English to a different language entirely: Some words will be represented by different arbitrary symbols.

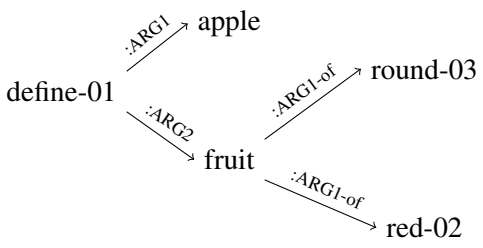


Figure 1: AMR obtained from translating the sentences "apple is defined as a red round fruit" and "a red round fruit is the definition of apple". As these sentences are definitions, *define-01* is the focus of the AMR. In Propbank, the ARG1 of define-01 is the *thing defined* and the ARG2 is the *definition*.

1. "s is defined as d."
2. "The definition of s is d."
3. "s has for definition d".
4. "d is the definition of s."
5. "s is defined by d."
6. "s gets defined as d."

Table 1: Some possible rephrasing of definitions, highlighting the definitional relation.

The second point of interest for embedding definitions in AMR is enrichment of the structure via a less atomic representation of meaning. For example while a word like *negation* is defined in a dictionary, AMR represents the concept of *negation* using polarity, or the notion of *conjunction* through instances like *or*. This can also be described as moving from a less strictly *lexicographic* approach to identifying *grounding sets* to a more *semantic* one.

3 From Dictionaries to AMR Digraphs

We now describe the process of embedding definitions of digital dictionaries into AMR. Subsection 3.1 introduces the desired structure when translating a definition into AMR. Subsection 3.2 in turn describes how AMR is leveraged to address polysemy. Subsection 3.3 completes the construction of AMR digraphs.

3.1 AMR Definitional Digraphs

The first step to embed the content of a dictionary into the AMR formalism is to create an AMR digraph out of each definition from the dictionary. In particular, such an approach needs to capture the *definitional* relation between the *defining words* and the *defined word*. In that spirit, one might

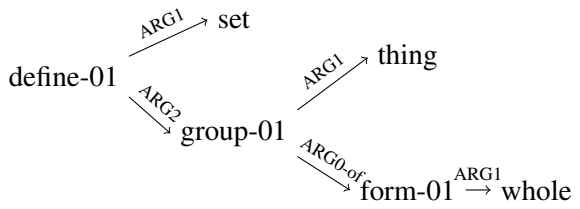


Figure 2: AMR generated from the sentence "set is defined as a group of things that form a whole".

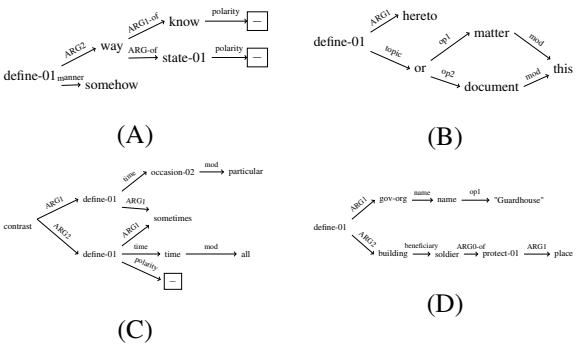


Figure 3: (A) An AMR without an ARG1. The arc here is mistakenly labeled as *manner*. (B) An AMR without an ARG2. The arc here is mistakenly labeled as *topic*. (C) An AMR with an undesired root node. The model interpreted the definition as a contrast between two definitions. (D) An AMR with a symbol defined by a subgraph.

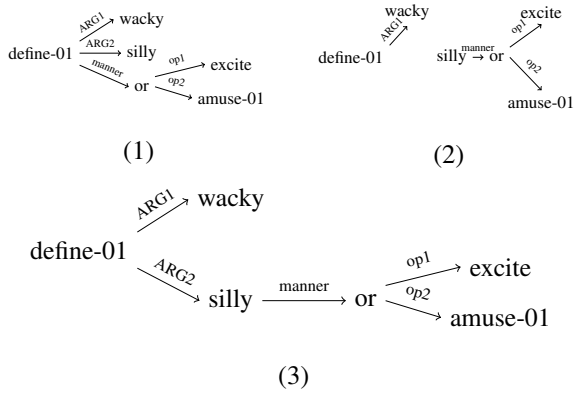


Figure 4: Process of correcting an invalid AMR digraph obtained by SToG for the sentence "wacky is defined as silly in an exciting or amusing way". **(1)** An invalid AMR digraph is detected. **(2)** Everything but the valid parts are deleted ; a new AMR digraph is generated using only the definition ("silly in an exciting or amusing way"). **(3)** The AMR digraph of the definition is bound to the root label by an ARG2 labeled arc.

rephrase each definition according to any of the sentences given in Table 1, where s is the defined symbol and d is the given definition of symbol s. Such sentences can then be translated into AMR embeddings using a state-of-the-art sentence-to-graph (StoG) parser (Jascob, 2023). In most cases, the resulting AMR digraph has the structure shown in Figures 1 and 2, i.e. the root of the graph is "define-01", its first argument is s and its second argument points to an AMR digraph of d. An AMR digraph having this structure is called *valid*.

Unfortunately, in many cases, the resulting graph does not exhibit the desired structure. Figure 3 provides examples of such *invalid* cases. If none of the rephrasing proposed in Table 1 is able to yield a valid AMR digraph, in some cases, it is possible to partially mitigate the loss by patching the resulting digraph. More specifically, assume that the produced AMR digraph for symbol s and definition d has a valid root node and has a single node designated by ARG1, which is s. Then it is possible to delete everything but the root and s, generate a new AMR for d, and then bind to ARG2 the root of the newly generated AMR digraph, to obtain a valid AMR digraph. This process is illustrated in Figure 4.

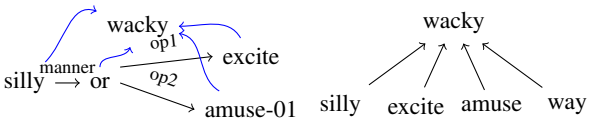
3.2 Handling Polysemy

Once the number of valid AMRs has been maximized, there remains the issue of *polysemy*: There can be many AMR definitional digraphs having the same label for s, since there is no guarantee that the polysemy of the processed digital dictionary is aligned with the polysemy represented in the AMR space.

In that case, we can rely on a strategy similar to that used in (Vincent-Lamarre et al., 2016), which consists in keeping only the first meaning with respect to the order used in the source digital dictionary. This process must be extended across all symbols to take into account collisions, as AMR might translate different words using the same label, and to prevent alteration of the structure of the dictionary.

3.3 AMR Digraphs

To finalize the transformation of the AMR digraphs, it remains to remove the AMR arcs and add arcs in a way that the definitional relation is correctly reflected. Since at that point each AMR is rooted with the label *define-01*, has a single AMR node to represent s and has a single subgraph to represent d, it can be assumed that each node in the definitional



(A) (B)

Figure 5: **(A)** An AMR where the root label has been bypassed. Each blue arc has been added after this bypass and is labelled *define-01* to signify a definitional relationship. **(B)** is the previous definitional graph format used in (Vincent-Lamarre et al., 2016). Both are made from the definition of "wacky".

subgraph is used to define the word at the end of ARG1. The root node can then be safely deleted and an outgoing arc labeled *define-01* is added to each node of the definitional subgraph towards the defined word.

This process is necessary to align the AMR formalism with the previously described approach to the *symbol grounding problem*. Figure 5 illustrates the difference between the two definition graph modeling approaches. We claim that the arcs added via this bypass capture *definitional relationships* similar to those used in (Vincent-Lamarre et al., 2016). As consequence, we can benefit from the same graph theoretical representation to identify *grounding sets* from AMR digraphs. We also preserve the removed AMR arcs to be added back to the *grounding sets*. More formally, *AMR digraphs* of dictionaries are defined as the union of the individual AMR definitional digraphs created from the embedding of the definitions described in the last subsections. This union can be seen as the graphical representation of the contents of a digital dictionary translated into AMR embeddings.

4 Reducing AMR Digraphs

A common first step of many algorithms solving the minimum feedback vertex set problem consists in applying reductions to the original digraph, while preserving, in some sense, its MFVS. In this section, we recall some of those reductions and discuss their *confluence property*.

4.1 Digraph Reductions

Let $\mathcal{G}$ be the set of all finite digraphs. A map $T : \mathcal{G} \to \mathcal{G}$ is called a *reduction* if, for any $G, G' \in \mathcal{G}$, such that $G = (V, A)$, $G' = (V', A')$ and $G' = T(G)$, either $|V'| < |V|$, or $|V'| = |V|$ and $|A'| < |A|$. In other words, a reduction either remove at least one vertex or, in the case where no vertex is removed, it removes at least one arc. Let T be a reduction, $G, G' \in \mathcal{G}$ be such that $G' = T(G)$ and $U \subseteq V(G)$. We say that T is *MFVS-preserving for G, with respect to U* if $U \cap V' = \emptyset$ and, for any MFVS U' of G', the set $U' \cup U$ is a MFVS of G. The set U is called a *partially constructed solution from T*.

Many MFVS-preserving reductions have been proposed in the literature (Levy and Low, 1988; Lin and Jou, 2000; Lemaic, 2008; Kiesel and Schidler, 2022). We recall some of them below. Let $G = (V, A)$ be any directed graph, $u, v \in V$. We define the following predicates:

- $\ell(G, u)$ if $(u, u) \in A$;
- $c(G, U)$ if U is a diclique of G;
- $i(G, u)$ if $\neg\ell(G, u)$ and $c(G, N^-_G(u))$;
- $o(G, u)$ if $\neg\ell(G, u)$ and $c(G, N^+_G(u))$;
- $s(G, u, v)$ if $\neg\ell(G, v)$, $(u, v) \in A$, $(v, u) \in A$, $N^-_G(v) \subseteq N^-_G(u)$ and $N^+_G(v) \subseteq N^+_G(u)$;
- $p(G, u, v)$ if (u, v) is acyclic in $G - A^{\leftrightarrow}(G)$.
- $d(G, u, v)$ if $(u, v) \in A$ and any $vu - path$ in $G - A^{\leftrightarrow}(G)$ goes through a vertex in $N^-_{G-A^{\leftrightarrow}(G)}(v) \cup N^+_{G-A^{\leftrightarrow}(G)}(u)$.

Pointed reductions, i.e. reductions targeting a specific vertex or arc, can then be defined as follows:

$$\textsc{Loop}(G, u) = \begin{cases} G - u & \text{if } \ell(G, u); \\ G & \text{otherwise.} \end{cases}$$

$$\textsc{InClique}(G, u) = \begin{cases} G \circ u & \text{if } i(G, u); \\ G & \text{otherwise.} \end{cases}$$

$$\textsc{OutClique}(G, u) = \begin{cases} G \circ u & \text{if } o(G, u); \\ G & \text{otherwise.} \end{cases}$$

$$\textsc{Subset}(G, u, v) = \begin{cases} G - u & \text{if } s(G, u, v); \\ G & \text{otherwise.} \end{cases}$$

$$\textsc{Pie}(G, u, v) = \begin{cases} G - (u, v) & \text{if } p(G, u, v); \\ G & \text{otherwise.} \end{cases}$$

$$\textsc{Dome++}(G, u, v) = \begin{cases} G - (u, v) & \text{if } d(G, u, v); \\ G & \text{otherwise.} \end{cases}$$

The effect of the reductions InClique and Pie are illustrated respectively in Figures 6 and 7. Moreover, proofs that each of these reductions are MFVS-preserving can be found in (Levy and Low,

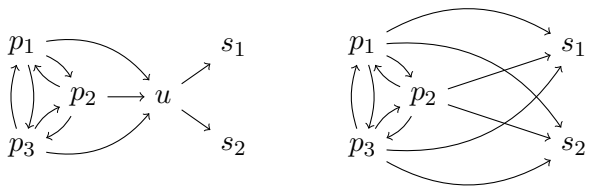


Figure 6: Illustration of the INCLIQUE reduction. Consider on the left the subgraph above of a digraph G where $i(G, u)$ is true, indeed $N_G^-(u) = \{p_1, p_2, p_3\}$ form a diclique. Then INCLIQUE(G, u) is applicable and on the right we see the result.

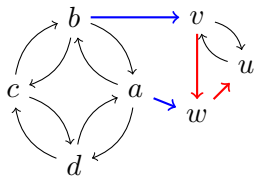


Figure 7: Illustration of the PIE reduction. Consider the digraph G above. Then PIE is applicable on the blue and red arcs. Indeed, there no circuit going through the blue arcs (b, v) and (a, w) in G. The same is true for the red arcs (v, w) and (w, u) in $G - A^{\leftrightarrow}(G)$. Therefore we can remove the blue and red arcs from G.

1988) for LOOP, in Lemaic's thesis for INCLIQUE and OUTCLIQUE (Lemaic, 2008), in Lin and Jou's paper for PIE (Lin and Jou, 2000), and in Kiesel and Schidler's paper for SUBSET and DOME++ (Kiesel and Schidler, 2022).

4.2 Confluence

Let G be a digraph and $\mathcal{R}$ be a set of reductions. We say that G is $\mathcal{R}$-*irreducible* if, for any $R \in \mathcal{R}$, $R(G) = G$. If there exists a sequence of reductions $R_1, R_2, \ldots, R_k \in \mathcal{R}$ such that $G' = (R_k \circ \cdots R_2 \circ R_1)(G)$, then we write $G \overset{\mathcal{R}}{\to} G'$ and say that G *reduces to* G'. If G' is $\mathcal{R}$-irreducible, then we write $G \overset{\mathcal{R}}{\Rightarrow} G'$ and then we say that G *fully reduces to* G'. Finally, we say that $\mathcal{R}$ is *confluent* if, for any digraph G, the conditions $G \overset{\mathcal{R}}{\Rightarrow} G'$ and $G \overset{\mathcal{R}}{\Rightarrow} G''$ imply $G' = G''$. Roughly speaking, confluence means that a digraph always fully reduces to the same graph, whatever the order in which we apply the reductions.

It was recently proved that the pointed operators LOOP$(\cdot,\cdot)$, INCLIQUE$(\cdot,\cdot)$, OUTCLIQUE$(\cdot,\cdot)$, SUBSET$(\cdot,\cdot,\cdot)$ and PIE$(\cdot,\cdot,\cdot)$ form a set of confluent reductions (Abdenbi et al., 2024). In particular, this means that the non pointed versions LOOP(G), INCLIQUE(G), OUTCLIQUE(G), SUBSET(G) and PIE(G) of the operators acting on all vertices or arcs yield the same result, whatever the order in which the vertices or the arcs are traveled. More generally, once the confluence of a set of reductions $\mathcal{R}$ is established, we are free to apply the confluent reductions in any order, with the guarantee that the final irreducible graph is unique up to isomorphism. From an algorithmic and computational complexity perspective, it is more efficient to apply the least costly reductions first, typically those with local applicability criteria, i.e., those involving a single vertex or arc and its neighborhood, such as LOOP or SUBSET. Afterward, we apply those that require more computation time, typically those with graph traversal-based criteria, such as PIE or DOME++.

Algorithm 1 Reduce with a set $\mathcal{R}$ of reductions

Input: G : digraph, $\mathcal{R}$: set of reductions, ρ : priority function
Output: an $\mathcal{R}$-*irreducible* digraph

```
function REDUCED(G, R, ρ)
    m ← max{ρ(R) | R ∈ R}
    return REDUCED(G, R, ρ, m)
function REDUCED(G, R, ρ, p)
    if p > 0 then
        G' ← G
        repeat
            G'' ← G'
            G'' ← REDUCED(G'', R, ρ, p−1)
            for R ∈ R do
                if ρ(R) = p then
                    G'' ← R(G'')
        until G' = G''
        return G'
    else
        return G
```

Algorithm 4.2 can be used to reduce a digraph with a given set of allowed reductions $\mathcal{R}$ whose order of application is driven by a priority function $\rho : \mathcal{R} \to \mathbb{N}_{>0}$ that associates with each reduction a positive integer (the smaller the value, the higher the priority). For instance, by setting $\mathcal{R}_c = \{$LOOP, SUBSET, INCLIQUE, OUTCLIQUE, PIE$\}$ and

$$\rho_c(R) = \begin{cases} 1, & \text{if } R \in \{\text{LOOP}, \text{SUBSET}, \\ & \quad \text{INCLIQUE}, \text{OUTCLIQUE}\}; \\ 2, & \text{if } R = \text{PIE}, \end{cases}$$

we obtain an efficient and confluent reduction algorithm A_c. Similarly, by setting $\mathcal{R}_{nc} = \mathcal{R}_c \cup$

$\{$DOME++$\}$, ρ_{nc}(DOME++) $= 3$ and $\rho_{nc}(R) = \rho_c(R)$ for $R \neq$ DOME++, we obtain another reduction algorithm A_{nc}, which has the potential to reduce the digraph further more, to the cost of sacrificing confluence.

5 Experimental Results

The section details the carried out experiment on real digital dictionaries.

We assembled 8 datasets coming from 5 different sources. Two of the dictionaries, the Longman's Dictionary of Contemporary English (LDOCE) (Procter, 1978), and the Cambridge International Dictionary of English (CIDE) (Procter, 1995), are built using a so-called *controlled vocabulary*, i.e. the words used in the definitions are limited as much as possible. LDOCE is an advanced learner's dictionary, originally published in 1978, while CIDE is a dictionary originally developed in 1995 for advanced learners of English using the Cambridge Corpus. The 3rd dictionary is the 11th edition of the Merriam-Webster's Collegiate Dictionary (MWC), published in 2003 (Merriam-Webster, 2003). The next dictionaries come from Wordsmyth (Wordsmyth, 2017), a linguistical educational project. They are divided into four specialized dictionaries: the Wordsmyth Educational Dictionary-Thesaurus (WEDT), first developed in 1980, followed later by the Wordsmyth Learner's Dictionary-Thesaurus (WLDT), the Wordsmyth Children's Dictionary-Thesaurus (WCDT) and the Wordsmyth Illustrated Learner's Dictionary (WILD). The first two are targeted at adults, WEDT being for advanced learners and WLDT for beginners, while the last two are aimed at children. Finally, WordNet (WN) (Fellbaum, 1998) is a well-known lexical network, whose purpose is not only to provide definitions of words, but also semantical relations between them, quasisynonymy, antonymy and hypernymy being the most important.

From these 8 dictionaries, we built 8 AMR digraphs using the ideas described in Section 3. Each symbol-definition pairs were translated into AMR using the pre-trained model with the best *smatch* (Cai and Knight, 2013) made available through AMRlib (Jascob, 2023). The *smatch* metric was proposed to evaluate the similarity between two AMR graphs by calculating the degree of overlap in their semantic structures. It is commonly used to measure the precision of AMR StoG models: Jascob (Jascob, 2023) reports *bart* as the best

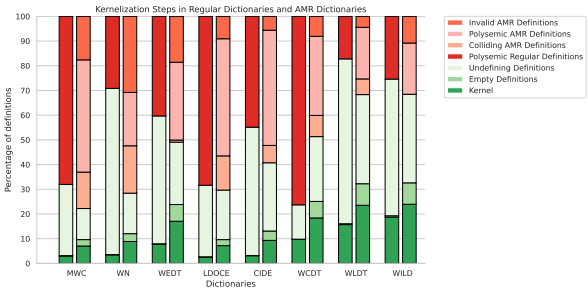


Figure 8: Shades of red in the upper part show number of definitions removed when starting with a complete digital dictionary and shades of green in the lower show the following preprocessing reducing it to its kernel. In each case, the left boxes show the values for the regular dictionaries, while the right boxes show the values for the AMR dictionaries.

pre-trained StoG model and mentions - without data - that for the time being, encoder-decoder architecture outperforms readily available decoder-only models (models like Llama and ChatGPT) for the task of translating English sentences into AMR. Inferring the AMR digraphs from hundreds of thousands of English sentences was a non-trivial task and were thus computed on the Narval supercomputer, a high-performance computing system provided by Compute Canada. All manipulations of AMR digraphs were done through the Penman library (Goodman, 2020) and Networkx (Hagberg et al., 2008).

In order to compare our results with those described in (Vincent-Lamarre et al., 2016), we also created digraphs from dictionaries using only the first definition of a given word. Only content words, i.e. open-class words, were considered. Each definition was transformed into a graph where words are nodes and definitional relationships are arcs. The incoming arcs of a given node correspond to the words used in its definition and the outgoing arcs of a given node point to words it defines. Figure 8 details the effect of applying those transformations for each of the 8 dictionaries.

Next, the reductions discussed in Section 4 were also applied on all 8 dictionaries. In order to have a more universal reduction representative, we applied the confluent version (A_c) as a first step and then we applied the nonconfluent version (A_{nc}). Figure 9 compares the proportion of application of each reduction, for each dictionary, with respect to regular dictionaries and AMR dictionaries, in the confluent case. Note that we are only representing the predominant vertex reductions (excluding SUBSET,

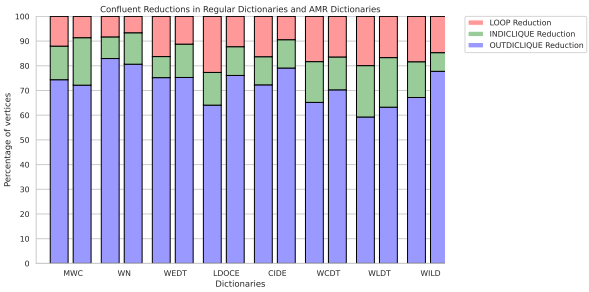


Figure 9: Proportion of applied confluent reductions in regular dictionaries (on the left) and AMR dictionaries (on the right).

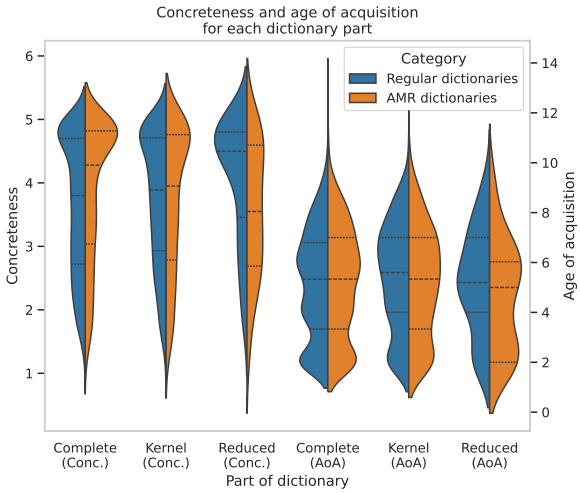


Figure 10: Distribution of the age of acquisition and concreteness for each type of dictionaries (regular and AMR) and each part of those dictionaries (complete, kernel and reduced).

as it is rarely applied). The focus is on reductions that affect the vertices of a dictionary or an AMR, because it is these latter which represent the *words*. Since PIE and DOME++ are reductions that only impact edges, they are not represented in Figure 9. Surprisingly, the majority of the reductions could be performed confluently. Hence, we did not include the non confluent reductions in Figure 9. All algorithms related to reduction of directed graph were implemented in C++. As both confluent and non confluent versions reduced all dictionaries in a matter of seconds or minutes, all computations were completed on a personal desktop.

Finally, we extracted all symbols or concepts common to all 8 digital dictionaries, for each dictionary part (complete, kernel and reduced) to study their psycholinguistical properties. We restricted our attention to two key psycholinguistic variables: *age of acquisition* (AOA) and *concreteness* (Conc). AOA refers to the average age at which a word is first learned, with data sourced from Brysbaert's database (covering ages 1 to 21) (Brysbaert and Biemiller, 2017). Concreteness measures the materiality of a word, ranging from *abstract* to *highly concrete*, based on participants' classifications. Examples include "banana" and "apple" at level 5, while "belief" is rated as level 1 (Brysbaert et al., 2014). Figure 10 illustrates the evolution of the distribution of both the age of acquisition and the concreteness across all dictionary parts.

6 Analysis and Discussion

First, it is worth mentioning that while embedding into AMR causes less definitions to be included in the graph, at the end, they result in larger kernels. This suggests that AMR digraphs are more able to capture semantic and definitional relations via a less atomic symbolic representation of meaning. Another notable observation is that the most applied reduction in all dictionaries, both AMR and regular, is OUTCLIQUE, as illustrated in Figure 9. This implies that many symbols or concepts used in some definitions can be replaced by others without losing the definitional path structure, the meaning of some implicit in others. That fact is consistent with the idea behind the CIDE and LDOCE dictionaries, aiming to reduce the used words to a smaller *controlled vocabulary*.

The violin plots illustrating the distribution of the age of acquisition and the concreteness yield interesting observations. The effect of the reduction on regular dictionaries is not obvious, while a clearer tendency can be observed on AMR dictionary parts: As we reduce the directed graphs, the concepts are learned at a younger age and they become more abstract. This is consistent with the intuition that words in the reduced digraph are more relevant to language learning.

7 Concluding Remarks

The main goal of this paper was to use the AMR formalism to study the symbol grounding problem. The experimental results obtained seem to show that AMR is more or less well suited to improve WSD when studying this problem, but that it leads to more interpretable digraph structure thanks to its larger *kernel*. Once minimal grounding sets are extracted from these, AMR arcs can be re-introduced to the remaining vertices, adding the notion of *what relations between these words need to be known*.

8 Limitations

A first limitation lies in the imposed priority in Algorithm 1. Although confluence guarantees a unique irreducible graph, the diagram in Figure 9 and Table 2 might still differ because INCLIQUE/OUTCLIQUE and PIE for example could overlap. Specifically, a vertex with a single predecessor (or successor) that does not belong to any circuit can be identified by both INCLIQUE (or OUTCLIQUE) and PIE, as all edges incident to this vertex are acyclic. From the perspective of studying the MFVS, it would be useful to exclude these vertices using PIE or DOME++ to assess how many vertices that appear in at least one circuit can be bypassed by INCLIQUE/OUTCLIQUE.

To a lesser extent, another limitation concerns the interpretability of the reductions used from a psycholinguistic perspective. Only the simpler versions of the INCLIQUE and OUTCLIQUE reductions can be interpreted when the vertex they apply to has at most one predecessor or successor. The removal of edges is mostly justified by the pursuit of the MFVS and lacks a clear psycholinguistic motivation. A more in-depth study would have been to assess, among the various reductions, how often each one applies individually to the initial dictionary or AMR.

Even though confluence ensures uniqueness, the order in which reductions are applied needs further exploration, not only from the perspective of graph theory and MFVS but also considering psycholinguistics and the impact of these reductions on the structure of a dictionary and an AMR.

Another limitation is the proportion of rejected definitions. Whilst we set out to alleviate the issues caused by polysemy and could leverage AMR in novel ways, the proportion of preserved definitions was lesser in all dictionaries. The non-atomicity of AMR embeddings of the defined word in a definition proved to be the most common cause of an invalid AMR. Having non-atomic labels that can easily be identified when used to define another word makes it almost impossible to recreate the definitional relationships. This being said, the principal reason for rejecting definitions was the collision of labels. Another issue lies in the limited number of symbols : AMR cannot contain all the possible definitions for all symbols included in the set of all dictionaries definitions.

References

Moussa Abdenbi, Alexandre Blondin Massé, Alain Goupil, and Odile Marcotte. 2024. On the confluence of directed graph reductions preserving feedback vertex set minimality. *Electronic Proceedings in Theoretical Computer Science*, 403:6–11.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Alexandre Blondin Massé, Guillaume Chicoisne, Yassine Gargouri, Stevan Harnad, Olivier Picard, and Odile Marcotte. 2008. How is meaning grounded in dictionary definitions? *arXiv preprint arXiv:0806.3710*.

Claire Bonial, Julie Foresta, Nicholas C Fung, Cory Hayes, Philip Osteen, Jacob Arkin, Benned Hedegaard, and Thomas Howard. 2023. Abstract meaning representation for grounded human-robot communication. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 34–44.

Marc Brysbaert and Andrew Biemiller. 2017. Test-based age-of-acquisition norms for 44 thousand english word meanings. *Behavior research methods*, 49:1520–1523.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46:904–911.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752.

Shibhansh Dohare, Harish Karnick, and Vivek Gupta. 2017. Text summarization using abstract meaning representation. *arXiv preprint arXiv:1706.01678*.

Christiane Fellbaum. 1998. Wordnet: An electronic lexical database.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for amr graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319.

Aric Hagberg, Pieter J Swart, and Daniel A Schult. 2008. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Stevan Harnad. 2024. Language writ large: Llms, chatgpt, grounding, meaning and understanding. *arXiv preprint arXiv:2402.02243*.

Benjamin Jascob. 2023. Amrlib models [software]. https://github.com/bjascob/amrlib-models. GitHub repository.

Richard M Karp. 2010. *Reducibility among combinatorial problems*. Springer.

Rafael Kiesel and André Schidler. 2022. A dynamic maxsat-based approach to directed feedback vertex sets. *Preprint*, arXiv:2211.06109.

Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. 2024. Text summarization based on semantic graphs: An abstract meaning representation graph-to-text deep learning approach. *Journal of Big Data*, 11(1):95.

Mile Lemaic. 2008. *Markov-Chain-Based Heuristics for the Feedback Vertex Set Problem for Digraphs*. Ph.D. thesis, Universität zu Köln.

Hanoch Levy and David W Low. 1988. A contraction algorithm for finding small cycle cutsets. *Journal of Algorithms*, 9(4):470–493.

Changmao Li and Jeffrey Flanigan. 2022. Improving neural machine translation with the abstract meaning representation by combining graph and sequence transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing*, pages 12–21. Association for Computational Linguistics.

Hen-Ming Lin and Jing-Yang Jou. 2000. On computing the minimum feedback vertex set of a directed graph by contraction operations. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 19:295 – 307.

Merriam-Webster. 2003. *Merriam-Webster's Collegiate Dictionary*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

P. Procter. 1978. *Longman Dictionary of Contemporary English (LDOCE)*.

P. Procter. 1995. *Cambridge International Dictionary of English (CIDE)*.

Nasim Tohidi and Chitra Dadkhah. 2022. A short review of abstract meaning representation applications. *Modeling and Simulation in Electrical and Electronics Engineering*, 2(3):1–9.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Philippe Vincent-Lamarre, Alexandre Blondin Massé, Marcos Lopes, Mélanie Lord, Odile Marcotte, and Stevan Harnad. 2016. The latent structure of dictionaries. *Topics in cognitive science*, 8(3):625–659.

Wordsmyth. 2017. Wordsmyth.

A Results Tables

This appendix section provides various results tables.

Table 2 presents the detailed results of the applications of different confluent reductions on dictionaries (white columns) and AMRs (red columns). Below are the details of each row:

- Nb Remain. The number of vertices remaining in the irreducible graph.
- Nb Incl. The number of vertices included in the partial MFVS.
- Nb Excl. The number of excluded vertices.
- Nb Reduc. The total number of reductions applied.
- Nb R. The number of times the reduction R was applied.
- Nb ISOL. The number of vertices whose incident edges have all been removed by PIE or DOME++.

Like Table 2, Table 3 presents the detailed results of the applications of different non-confluent reductions on dictionaries (white columns) and AMRs (red columns). The rows have the same meaning as in Table 2.

Table 4 summarizes the metrics of the graphs created from the union of graphs made from individual definitions. Once again, regular dictionaries metrics are shown in white columns and AMR's in red columns. While it is true that less definitions are taken into account by AMR, it still has a somewhat more dense representation of meaning as even when taking into account the loss in definitions. It is also noteworthy that AMR embedding have a much smaller proportions of undefining words, implying what words appear are generally more often used. It also has a far higher proportion of undefined words : this is due to the fact that AMR is like translating from English to another dictionary. Many used symbols are left undefined as they are lifted from the Propbank Frames. This suggests a potential solution would be to source missing definitions in Propbank Frames. The kernels of regular

	WordNet		MerWeb		WEDT		LDOCE		CIDE		WCDT		WLDT		WILD	
Nb Remain.	2 068	2 948	1 968	2 846	1 601	1 836	436	931	696	1 067	389	374	151	95	408	216
Nb Incl.	548	4 697	1 125	6 233	899	4 465	390	1 411	185	1 643	460	1 591	208	692	172	457
Nb Excl.	143 882	44 508	94 402	50 332	50 065	26 349	34 773	19 527	20 214	16 202	11 530	8 655	4 149	3 688	2 633	2 391
Nb Reduc.	5 120	3 969	7 081	4 879	5 293	3 386	1 576	1 291	1 055	998	2 531	1 545	966	812	982	463
Nb LOOP	409	258	818	417	814	360	341	155	162	93	415	253	185	133	160	68
Nb SUBSET	3	1	2	0	3	0	3	0	2	0	4	2	1	1	1	0
Nb IN	431	495	924	929	427	433	199	147	113	113	374	205	193	160	126	35
Nb OUT	4 084	3 137	5 051	3 487	3 763	2 414	963	961	717	779	1 477	1 082	550	504	585	360
Nb PIE	193	78	284	46	282	179	70	28	61	13	259	3	36	14	110	0
Nb ISOL.	0	0	2	0	4	0	0	0	0	0	0	0	1	0	0	0

Table 2: Confluent reductions metrics. Left column: regular dictionaries. Right column: AMR dictionaries.

	WordNet		MerWeb		WEDT		LDOCE		CIDE		WCDT		WLDT		WILD	
Nb Remain.	2 028	2 928	1 933	2 807	1 556	1 822	423	915	683	1 058	342	353	127	85	398	184
Nb Incl.	552	4 697	1 127	6 234	907	4 466	394	1 412	187	1 644	466	1 594	213	693	173	466
Nb Excl.	143 918	44 528	94 435	50 370	50 102	26 362	34 782	19 542	20 225	16 210	11 571	8 673	4 168	3 697	2 642	2 414
Nb Reduc.	5 498	4 481	7 526	5 421	5 733	3 653	1 727	1 488	1 184	1 253	2 828	1 721	1 086	879	1 180	827
Nb LOOP	412	258	820	418	820	361	345	156	164	93	420	256	185	134	161	77
Nb SUBSET	4	1	2	0	5	0	3	0	2	1	7	2	1	1	1	0
Nb IN	457	504	940	945	441	442	205	155	118	117	394	185	206	111	133	53
Nb OUT	4 093	3 146	5 067	3 509	3 789	2 418	966	968	723	783	1 497	1 085	556	607	587	364
Nb PIE	193	78	284	46	298	179	70	28	61	13	259	3	36	60	110	4
Nb DOME++	338	492	410	503	379	253	138	181	116	246	250	155	96	165	188	328
Nb ISOL.	1	2	3	0	1	0	0	0	0	0	1	0	1	0	0	1

Table 3: Non-confluent reductions metrics. Left column: regular dictionaries. Right column: AMR dictionaries.

dictionary are also generally almost twice as dense as those of AMR dictionaries. Here are the details of each row :

- Nb Vertices : The total number of vertices in the graph before reductions
- Size Kernel The number of vertices remaining once all non-defined and non-defining words have been recursively removes
- Size Red. Kernel : The number of vertices in the reduced kernel and is equivalent to the remaining vertices in the confluent reductions table.
- NC Red. Kernel : The number of vertices in the non-confluently reduced kernel. Is equivalent to the remaining vertices in the non-confluent reductions tables.
- Init Nb Arcs : The initial number of arcs in the graph before reductions
- Final Nb Arcs : The number of arcs after confluent reductions
- NC Final Arcs : The number of arcs after non-confluent reductions
- Nb Undefined : The number of words removed when recursively removing words not defined by any other
- Nb Undefining : The number of words removed when recursively removing all words not used to define another.
- Nb SCCs Kernel : The number of strongly connected components found in the kernel. All kernels are made up almost entirely of their biggest strongly connected component and all other SCCs are of negligible size.
- Kernel Nb Arcs : The number of arcs for a given kernel
- Kernel Density : The density of each kernel.

Table 5 describes the metrics of creating AMR graphs from definitions. It is noteworthy that the principal reason for discarding a definition embedded in AMR is label collision, within a symbol's definitions and across symbols definitions. Here are the details for each row :

- Definition Quantity : The total number of definitions in a dictionary, including polysemic definitions (when a word has multiple definitions).
- Initial Invalid Quantity : The initial quantity of AMR graphs that were not valid as per our requirements described in 3
- Save Quantity : The number of AMR graphs saved by making them *valid* via our two strategies described in 3.1.
- Final Invalid Quantity : The number of graphs that could not be made valid.
- Polysemy Filtered : The number of polysemic definitions lost due to having the same label for the defined word.

	WordNet		MerWeb		WEDT		LDOCE		CIDE		WCDT		WLDT		WILD	
Nb vertices	146 498	52 153	97 495	59 411	52 565	32 650	35 599	21 869	21 095	18 912	12 379	10 620	4 508	4 475	3 213	3 064
Size Kernel	7 131	11 277	9 070	13 495	6 694	9 148	1 988	3 450	1 711	3 602	2 813	3 252	1 103	1 451	1 291	1 068
Size Red. Ker.	2 068	2 948	1 968	2 846	1 601	1 836	436	931	696	1 067	389	374	147	95	408	216
NC Red Ker.	2 028	2 928	1 933	2 807	1 556	1 822	423	915	683	1 058	342	353	123	85	398	184
Init. Nb Arcs	765 481	311321	495 385	337 376	273 432	190 470	173 737	147 319	107 638	144 308	61 282	56 102	19 133	21 683	22 667	16 579
Final Nb Arcs	14 858	32 277	14 278	33 268	11 109	19 147	2 700	8 943	4 837	11 940	2 524	3 665	833	1 149	2 860	1 828
NC Final Arcs	14 430	31 766	13 831	32 754	10 520	18 849	2 539	8 741	4 697	11 660	2 177	3 460	703	1 076	2 664	1 424
Nb Undefined	144	4 504	328	6 010	89	4 309	46	1 279	22	1 565	48	1 365	33	573	11	389
Nb Undefining	137 815	40 509	86 374	44 626	44 280	22 863	33 246	18 261	18 843	15 105	9 278	7 316	3 261	2997	1 874	1 991
Nb SCCs Kernel	253	81	459	188	385	112	168	71	65	23	125	68	49	27	18	22
Kernel Nb Arcs	33 284	50 620	46 405	58 260	33 580	35 308	10 014	16 058	9 792	17 459	13 929	10 295	4 459	4 306	8 618	3 728
Kernel Density	0.0007	0.0004	0.0006	0.0003	0.0007	0.0004	0.0025	0.0014	0.0034	0.0014	0.0018	0.0010	0.0037	0.0020	0.0052	0.0033

Table 4: Graph metrics. Left column: regular dictionaries. Right column: AMR dictionaries.

	MerWeb	WordNet	WEDT	LDOCE	CIDE	WCDT	WLDT	WILD
Definition Quantity	301 240	206 185	86 949	80 086	49 787	22 563	6 900	4 709
Initial Invalid Quantity	98 349	98 443	28 548	17 030	10 484	3 910	739	796
Saved Quantity	43 632	32 195	13 721	9 538	7 564	1 936	400	231
Final Invalid Quantity	54 717	66 248	14 827	7 492	2 920	1 974	339	565
Polysemy Filtered	141 255	46 486	25 137	39 257	24 296	7 822	1 600	1 085
Symbols Collisions	45 587	41 298	14 335	11 468	3 659	2 097	486	5
Final Quantity	59 411	52 153	32 650	21 869	18 912	10 620	4 475	3 064

Table 5: AMR digraphs preprocessing metrics.

animal	fight	high	paper	thread
attention	finger	law	picture	time
bird	flat	leg	plant	touch
building	food	letter	power	train
control	foot	life	protect	tree
curve	game	liquid	room	verb
decide	gas	machine	season	water
earth	glass	metal	sleep	wheel
eat	grain	mind	soft	wing
end	hand	money	solid	word
face	happen	new	sun	work
farm	head	number	taste	write
female				

Table 6: List of regular dictionary symbols common to the 8 digital reduced dictionaries

- Symbols Collisions : The number of definitions lost due to having the same label for the label for the defined word across different words.
- Final Quantity : The final quantity of definitions used in the graph.

Table 6 gives the list of the symbols common to the 8 reduced regular dictionaries. Similarly, Table 7 gives the AMR concepts occurring in each of the 8 reduced AMR dictionaries.

another	have-degree-91	position
area	have-frequency-91	product-of
ask-02	have-org-role-91	real-04
at-a-time	have-part-91	regular-03
be-located-at-91	have-quant-91	relative-position
brown-01	have-rel-role-91	responsible-03
brown-02	information	return-05
building	in-front-of	right
careful	join-01	road
catch-01	keep-04	round-06
clothes	know-02	rule
come-03	leg	same-01
complete-02	make-05	send-03
concern-02	marry-01	size-01
copy-01	mass-quantity	smooth-06
country	member	son
cover-02	mind	space
cut-01	most	speak-01
danger	mouth	spend-02
dark-03	new-01	stiff-04
date-entity	number	string-entity
date-interval	officer	sweet-04
decide-01	one	temporal-quantity
distance-quantity	only	that
dry-08	open-06	thing
energy	open-09	together
enough	opening	top
equal-01	ordinal-entity	touch-01
every	part	use-03
feel-01	particular	value-02
fit-06	person	water
food	picture	white-03
force-01	piece	whole
force-04	place-01	word
get-04	planet	work-09
good-03	plant	write-01
grass	play-02	wrong-02
group	play-11	yellow-02
happy-02		

Table 7: List of AMR concepts common to the 8 digital reduced dictionaries