Abstract

We describe Abstract Meaning Representation (AMR), a semantic representation language in which we are writing down the meanings of thousands of English sentences. We hope that a sembank of simple, whole-sentence semantic structures will spur new work in statistical natural language understanding and generation, like the Penn Treebank encouraged work on statistical parsing. This paper gives an overview of AMR and tools associated with it.

1 Introduction

Syntactic treebanks have had tremendous impact on natural language processing. The Penn Treebank is a classic example—a simple, readable file of natural-language sentences paired with rooted, labeled syntactic trees. Researchers have exploited manually-built treebanks to build statistical parsers that improve in accuracy every year. This success is due in part to the fact that we have a single, whole-sentence parsing task, rather than separate tasks and evaluations for base noun identification, prepositional phrase attachment, trace recovery, verb-argument dependencies, etc. Those smaller tasks are naturally solved as a by-product of whole-sentence parsing, and in fact, solved better than when approached in isolation.

By contrast, semantic annotation today is balkanized. We have separate annotations for named entities, co-reference, semantic relations, discourse connectives, temporal entities, etc. Each annotation has its own associated evaluation, and training data is split across many resources. We lack a simple readable sembank of English sentences paired with their whole-sentence, logical meanings. We believe a sizable sembank will lead to new work in statistical natural language understanding (NLU), resulting in semantic parsers that are as ubiquitous as syntactic ones, and support natural language generation (NLG) by providing a logical semantic input.

Of course, when it comes to whole-sentence semantic representations, linguistic and philosophical work is extensive. We draw on this work to design an Abstract Meaning Representation (AMR) appropriate for sembanking. Our basic principles are:

- AMRs are rooted, labeled graphs that are easy for people to read, and easy for programs to traverse.
- AMR aims to abstract away from syntactic idiosyncrasies. We attempt to assign the same AMR to sentences that have the same basic meaning. For example, the sentences "he described her as a genius", "his description of her: genius", and "she was a genius, according to his description" are all assigned the same AMR.
- AMR makes extensive use of PropBank framesets (Kingsbury and Palmer, 2002; Palmer et al., 2005). For example, we represent a phrase like "bond investor" using the frame "invest-01", even though no verbs appear in the phrase.
- AMR is agnostic about how we might want to derive meanings from strings, or vice-versa. In translating sentences to AMR, we do not dictate a particular sequence of rule applications or provide alignments that reflect such rule sequences. This makes sembanking very fast, and it allows researchers to explore their own ideas about how strings

are related to meanings.

- AMR is heavily biased towards English. It is not an Interlingua.

AMR is described in a 50-page annotation guideline.[1] In this paper, we give a high-level description of AMR, with examples, and we also provide pointers to software tools for evaluation and sembanking.

2 AMR Format

We write down AMRs as rooted, directed, edge-labeled, leaf-labeled graphs. This is a completely traditional format, equivalent to the simplest forms of feature structures (Shieber et al., 1986), conjunctions of logical triples, directed graphs, and PENMAN inputs (Matthiessen and Bateman, 1991). Figure 1 shows some of these views for the sentence “The boy wants to go”. We use the graph notation for computer processing, and we adapt the PENMAN notation for human reading and writing.

3 AMR Content

In neo-Davidsonian fashion (Davidson, 1969), we introduce variables (or graph nodes) for entities, events, properties, and states. Leaves are labeled with concepts, so that “(b / boy)” refers to an instance (called b) of the concept boy. Relations link entities, so that “(d / die-01 :location (p / park))” means there was a death (d) in the park (p). When an entity plays multiple roles in a sentence, we employ re-entrancy in graph notation (nodes with multiple parents) or variable re-use in PENMAN notation.

AMR concepts are either English words (“boy”), PropBank framesets (“want-01”), or special keywords. Keywords include special entity types (“date-entity”, “world-region”, etc.), quantities (“monetary-quantity”, “distance-quantity”, etc.), and logical conjunctions (“and”, etc).

AMR uses approximately 100 relations:

- **Frame arguments, following PropBank conventions.** :arg0, :arg1, :arg2, :arg3, :arg4, :arg5.

- **General semantic relations.** :accompanier, :age, :beneficiary, :cause, :compared-to, :concession, :condition, :consist-of, :degree, :destination, :direction, :domain, :duration,

[1] AMR guideline: amr.isi.edu/language.html

LOGIC format:

∃ w, b, g:
instance(w, want-01) ∧ instance(g, go-01) ∧
instance(b, boy) ∧ arg0(w, b) ∧
arg1(w, g) ∧ arg0(g, b)

AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

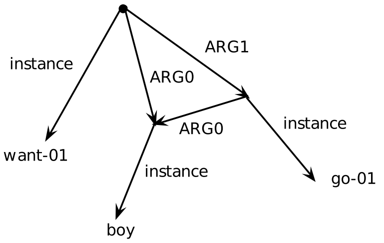

Figure 1: Equivalent formats for representating the meaning of "The boy wants to go".

:employed-by, :example, :extent, :frequency, :instrument, :li, :location, :manner, :medium, :mod, :mode, :name, :part, :path, :polarity, :poss, :purpose, :source, :subevent, :subset, :time, :topic, :value.

- **Relations for quantities.** :quant, :unit, :scale.
- **Relations for date-entities.** :day, :month, :year, :weekday, :time, :timezone, :quarter, :dayperiod, :season, :year2, :decade, :century, :calendar, :era.
- **Relations for lists.** :op1, :op2, :op3, :op4, :op5, :op6, :op7, :op8, :op9, :op10.

AMR also includes the inverses of all these relations, e.g., :arg0-of, :location-of, and :quant-of. In addition, every relation has an associated reification, which is what we use when we want to modify the relation itself. For example, the reification of :location is the concept "be-located-at-91".

Our set of concepts and relations is designed to allow us represent all sentences, taking all words into account, in a reasonably consistent manner. In the rest of this section, we give examples of how AMR represents various kinds of words, phrases, and sentences. For full documentation, the reader is referred to the AMR guidelines.

Frame arguments. We make heavy use of PropBank framesets to abstract away from English syntax. For example, the frameset "describe-01" has three pre-defined slots (:arg0 is the describer, :arg1 is the thing described, and :arg2 is what it is being described as).

```
(d / describe-01
  :arg0 (m / man)
  :arg1 (m2 / mission)
  :arg2 (d / disaster))

The man described the mission as a disaster.
The man's description of the mission:
  disaster.
As the man described it, the mission was a
  disaster.
```

Here, we do not annotate words like "as" or "it", considering them to be syntactic sugar.

General semantic relations. AMR also includes many non-core relations, such as :beneficiary, :time, and :destination.

```
(s / hum-02
  :arg0 (s2 / soldier)
  :beneficiary (g / girl)
  :time (w / walk-01
          :arg0 g
          :destination (t / town)))

The soldier hummed to the girl as she
  walked to town.
```

Co-reference. AMR abstracts away from co-reference gadgets like pronouns, zero-pronouns, reflexives, control structures, etc. Instead we re-use AMR variables, as with "g" above. AMR annotates sentences independent of context, so if a pronoun has no antecedent in the sentence, its nominative form is used, e.g., "(h / he)".

Inverse relations. We obtain rooted structures by using inverse relations like :arg0-of and :quant-of.

```
(s / sing-01
  :arg0 (b / boy
          :source (c / college)))

The boy from the college sang.

(b / boy
  :arg0-of (s / sing-01)
  :source (c / college))

the college boy who sang ...

(i / increase-01
  :arg1 (n / number
          :quant-of (p / panda)))

The number of pandas increased.
```

The top-level root of an AMR represents the focus of the sentence or phrase. Once we have selected the root concept for an entire AMR, there are no more focus considerations—everything else is driven strictly by semantic relations.

Modals and negation. AMR represents negation logically with :polarity, and it expresses modals with concepts.

```
(g / go-01
  :arg0 (b / boy)
  :polarity -)

The boy did not go.

(p / possible
  :domain (g / go-01
            :arg0 (b / boy))
  :polarity -))

The boy cannot go.
It's not possible for the boy to go.

(p / possible
  :domain (g / go-01
            :arg0 (b / boy)
            :polarity -))

It's possible for the boy not to go.

(p / obligate-01
  :arg2 (g / go-01
          :arg0 (b / boy))
  :polarity -)

The boy doesn't have to go.
The boy isn't obligated to go.
The boy need not go.

(p / obligate-01
  :arg2 (g / go-01
          :arg0 (b / boy)
          :polarity -))

The boy must not go.
It's obligatory that the boy not go.

(t / think-01
  :arg0 (b / boy)
  :arg1 (w / win-01
          :arg0 (t / team)
          :polarity -))

The boy doesn't think the team will win.
The boy thinks the team won't win.
```

Questions. AMR uses the concept “amr-unknown”, in place, to indicate wh-questions.

```
(f / find-01
  :arg0 (g / girl)
  :arg1 (a / amr-unknown))

What did the girl find?

(f / find-01
  :arg0 (g / girl)
  :arg1 (b / boy)
  :location (a / amr-unknown))

Where did the girl find the boy?
```

```
(f / find-01
  :arg0 (g / girl)
  :arg1 (t / toy
          :poss (a / amr-unknown)))

Whose toy did the girl find?
```

Yes-no questions, imperatives, and embedded wh-clauses are treated separately with the AMR relation :mode.

Verbs. Nearly every English verb and verb-particle construction we have encountered has a corresponding PropBank frameset.

```
(l / look-05
  :arg0 (b / boy)
  :arg1 (a / answer))

The boy looked up the answer.
The boy looked the answer up.
```

AMR abstracts away from light-verb constructions.

```
(a / adjust-01
  :arg0 (g / girl)
  :arg1 (m / machine))

The girl adjusted the machine.
The girl made adjustments to the machine.
```

Nouns. We use PropBank verb framesets to represent many nouns as well.

```
(d / destroy-01
  :arg0 (b / boy)
  :arg1 (r / room))

the destruction of the room by the boy ...
the boy's destruction of the room ...
The boy destroyed the room.
```

We never say "destruction-01" in AMR. Some nominalizations refer to a whole event, while others refer to a role player in an event.

```
(s / see-01
  :arg0 (j / judge)
  :arg1 (e / explode-01))

The judge saw the explosion.

(r / read-01
  :arg0 (j / judge)
  :arg1 (t / thing
          :arg1-of (p / propose-01))

The judge read the proposal.

(t / thing
  :arg1-of (o / opine-01
             :arg0 (g / girl)))

the girl's opinion
the opinion of the girl
what the girl opined
```

Many "-er" nouns invoke PropBank framesets. This enables us to make use of slots defined for those framesets.

```
(p / person
  :arg0-of (i / invest-01))

investor

(p / person
  :arg0-of (i / invest-01
             :arg1 (b / bond)))

bond investor

(p / person
  :arg0-of (i / invest-01
             :manner (s / small)))

small investor

(w / work-01
  :arg0 (b / boy)
  :manner (h / hard))

the boy is a hard worker
the boy works hard
```

However, a treasurer is not someone who treasures, and a president is not (just) someone who presides.

Adjectives. Various adjectives invoke PropBank framesets.

```
(s / spy
  :arg0-of (a / attract-01))

the attractive spy

(s / spy
  :arg0-of (a / attract-01
             :arg1 (w / woman)))

the spy who is attractive to women
```

"-ed" adjectives frequently invoke verb framesets. For example, "acquainted with magic" maps to "acquaint-01". However, we are not restricted to framesets that can be reached through morphological simplification.

```
(f / fear-01
  :arg0 (s / soldier)
  :arg1 (b / battle-01))

The soldier was afraid of battle.
The soldier feared battle.
The soldier had a fear of battle.
```

For other adjectives, we have defined new framesets.

```
(r / responsible-41
  :arg1 (b / boy)
  :arg2 (w / work))

The boy is responsible for the work.
The boy has responsibility for the work.
```

While "the boy responsibles the work" is not good English, it is perfectly good Chinese. Similarly, we handle *tough*-constructions logically.

```
(t / tough
  :domain (p / please-01
             :arg1 (g / girl)))

Girls are tough to please.
It is tough to please girls.
Pleasing girls is tough.
```

"please-01" and "girl" are adjacent in the AMR, even if they are not adjacent in English. "-able" adjectives often invoke the AMR concept "possible", but not always (e.g., a "taxable fund" is actually a "taxed fund").

```
(s / sandwich
  :arg1-of (e / eat-01
              :domain-of (p / possible)))

an edible sandwich

(f / fund
  :arg1-of (t / tax-01))

a taxable fund
```

Pertainym adjectives are normalized to root form.

```
(b / bomb
  :mod (a / atom))

atom bomb
atomic bomb
```

Prepositions. Most prepositions simply signal semantic frame elements, and are themselves dropped from AMR.

```
(d / default-01
  :arg1 (n / nation)
  :time (d2 / date-entity
          :month 6))

The nation defaulted in June.
```

Time and location prepositions are kept if they carry additional information.

```
(d / default-01
  :arg1 (n / nation)
  :time (a / after
          :op1 (w / war-01)))

The nation defaulted after the war.
```

Occasionally, neither PropBank nor AMR has an appropriate relation, in which case we hold our nose and use a :prep-X relation.

```
(s / sue-01
  :arg1 (m / man)
  :prep-in (c / case))

The man was sued in the case.
```

Named entities. Any concept in AMR can be modified with a :name relation. However, AMR includes standardized forms for approximately 80 named-entity types, including person, country, sports-facility, etc.

```
(p / person
  :name (n / name
          :op1 "Mollie"
          :op2 "Brown"))

Mollie Brown

(p / person
  :name (n / name
          :op1 "Mollie"
          :op2 "Brown")
  :arg0-of (s / slay-01
             :arg1 (o / orc)))

the orc-slaying Mollie Brown
Mollie Brown, who slew orcs
```

AMR does not normalize multiple ways of referring to the same concept (e.g., "US" versus "United States"). It also avoids analyzing semantic relations inside a named entity—e.g., an organization named "Stop Malaria Now" does not invoke the "stop-01" frameset. AMR gives a clean, uniform treatment to titles, appositives, and other constructions.

```
(c / city
  :name (n / name
          :op1 "Zintan"))

Zintan
the city of Zintan

(p / president
  :name (n / name
          :op1 "Obama"))

President Obama
Obama, the president ...

(g / group
  :name (n / name
          :op1 "Elsevier"
          :op2 "N.V.")
  :mod (c / country
         :name (n2 / name
                 :op1 "Netherlands"))
  :arg0-of (p / publish-01))

Elsevier N.V., the Dutch publishing group...
Dutch publishing group Elsevier N.V. ...
```

Copula. Copulas use the :domain relation.

```
(w / white
  :domain (m / marble))

The marble is white.

(l / lawyer
  :domain (w / woman))

The woman is a lawyer.

(a / appropriate
  :domain (c / comment)
  :polarity -))

The comment is not appropriate.
```

```
The comment is inappropriate.
```

Reification. Sometimes we want to use an AMR relation as a first-class concept—to be able to modify it, for example. Every AMR relation has a corresponding reification for this purpose.

```
(m / marble
  :location (j / jar))

the marble in the jar ...

(b / be-located-at-91
  :arg1 (m / marble)
  :arg2 (j / jar)
  :polarity -)
  :time (y / yesterday))

The marble was not in the jar yesterday.
```

If we do not use the reification, we run into trouble.

```
(m / marble
  :location (j / jar
              :polarity -)
  :time (y / yesterday))

yesterday's marble in the non-jar ...
```

Some reifications are standard PropBank framesets (e.g., "cause-01" for :cause, or "age-01" for :age).

This ends the summary of AMR content. For lack of space, we omit descriptions of comparatives, superlatives, conjunction, possession, determiners, date entities, numbers, approximate numbers, discourse connectives, and other phenomena covered in the full AMR guidelines.

4 Limitations of AMR

AMR does not represent inflectional morphology for tense and number, and it omits articles. This speeds up the annotation process, and we do not have a nice semantic target representation for these phenomena. A lightweight syntactic-style representation could be layered in, via an automatic post-process.

AMR has no universal quantifier. Words like "all" modify their head concepts. AMR does not distinguish between real events and hypothetical, future, or imagined ones. For example, in "the boy wants to go", the instances of "want-01" and "go-01" have the same status, even though the "go-01" may or may not happen.

We represent "history teacher" nicely as "(p / person :arg0-of (t / teach-01 :arg1 (h / history)))". However, "history professor" becomes "(p / professor :mod (h / history))", because "profess-01" is not an appropriate frame. It would be reasonable in such cases to use a NomBank (Meyers et al., 2004) noun frame with appropriate slots.

5 Creating AMRs

We have developed a power editor for AMR, accessible by web interface.[2] The AMR Editor allows rapid, incremental AMR construction via text commands and graphical buttons. It includes online documentation of relations, quantities, reifications, etc., with full examples. Users log in, and the editor records AMR activity. The editor also provides significant guidance aimed at increasing annotator consistency. For example, users are warned about incorrect relations, disconnected AMRs, words that have PropBank frames, etc. Users can also search existing sembanks for phrases to see how they were handled in the past. The editor also allows side-by-side comparison of AMRs from different users, for training purposes.

In order to assess inter-annotator agreement (IAA), as well as automatic AMR parsing accuracy, we developed the *smatch* metric (Cai and Knight, 2013) and associated script.[3] Smatch reports the semantic overlap between two AMRs by viewing each AMR as a conjunction of logical triples (see Figure 1). Smatch computes precision, recall, and F-score of one AMR's triples against the other's. To match up variables from two input AMRs, smatch needs to execute a brief search, looking for the variable mapping that yields the highest F-score.

Smatch makes no reference to English strings or word indices, as we do not enforce any particular string-to-meaning derivation. Instead, we compare semantic representations directly, in the same way that the MT metric Bleu (Papineni et al., 2002) compares target strings without making reference to the source.

For an initial IAA study, and prior to adjusting the AMR Editor to encourage consistency, 4 expert AMR annotators annotated 100 newswire sentences and 80 web text sentences. They then created consensus AMRs through discussion. The average annotator vs. consensus IAA (smatch) was 0.83 for newswire and 0.79 for web text. When newly trained annotators doubly annotated 382 web text sentences, their annotator vs. annotator IAA was 0.71.

[2] AMR Editor: amr.isi.edu/editor.html

[3] Smatch: amr.isi.edu/evaluation.html

6 Current AMR Bank

We currently have a manually-constructed AMR bank of several thousand sentences, a subset of which can be freely downloaded,[4] the rest being distributed via the LDC catalog.

In initially developing AMR, the authors built consensus AMRs for:

- 225 short sentences for tutorial purposes
- 142 sentences of newswire (*)
- 100 sentences of web data (*)

Trained annotators at LDC then produced AMRs for:

- 1546 sentences from the novel "The Little Prince"
- 1328 sentences of web data
- 1110 sentences of web data (*)
- 926 sentences from Xinhua news (*)
- 214 sentences from CCTV broadcast conversation (*)

Collections marked with a star (*) are also in the OntoNotes corpus (Pradhan et al., 2007; Weischedel et al., 2011).

Using the AMR Editor, annotators are able to translate a full sentence into AMR in 7-10 minutes and postedit an AMR in 1-3 minutes.

7 Related Work

Researchers working on whole-sentence semantic parsing today typically use small, domain-specific sembanks like GeoQuery (Wong and Mooney, 2006). The need for larger, broad-coverage sembanks has sparked several projects, including the Groningen Meaning Bank (GMB) (Basile et al., 2012a), UCCA (Abend and Rappoport, 2013), the Semantic Treebank (ST) (Butler and Yoshimoto, 2012), the Prague Dependency Treebank (Böhmová et al., 2003), and UNL (Uchida et al., 1999; Uchida et al., 1996; Martins, 2012).

Concepts. Most systems use English words as concepts. AMR uses PropBank frames (e.g., "describe-01"), and UNL uses English WordNet synsets (e.g., "200752493").

Relations. GMB uses VerbNet roles (Schuler, 2005), and AMR uses frame-specific PropBank relations. UNL has a dedicated set of over 30 frequently used relations.

Formalism. GMB meanings are written in DRT (Kamp et al., 2011), exploiting full first-

[4] amr.isi.edu/download.html

order logic. GMB and ST both include universal quantification.

Granularity. GMB and UCCA annotate short texts, so that the same entity can participate in events described in different sentences; other systems annotate individual sentences.

Entities. AMR uses 80 entity types, while GMB uses 7.

Manual versus automatic. AMR, UNL, and UCCA annotation is fully manual. GMB and ST produce meaning representations automatically, and these can be corrected by experts or crowds (Venhuizen et al., 2013).

Derivations. AMR and UNL remain agnostic about the relation between strings and their meanings, considering this a topic of open research. ST and GMB annotate words and phrases directly, recording derivations as (for example) Montague-style compositional semantic rules operating on CCG parses.

Top-down verus bottom-up. AMR annotators find it fast to construct meanings from the top down, starting with the main idea of the sentence (though the AMR Editor allows bottom-up construction). GMB and UCCA annotators work bottom-up.

Editors, guidelines, genres. These projects have graphical sembanking tools (e.g., Basile et al. (2012b)), annotation guidelines,[5] and sembanks that cover a wide range of genres, from news to fiction. UNL and AMR have both annotated many of the same sentences, providing the potential for direct comparison.

8 Future Work

Sembanking. Our main goal is to continue sembanking. We would like to employ a large sembank to create shared tasks for natural language understanding and generation. These tasks may additionally drive interest in theoretical frameworks for probabilistically mapping between graphs and strings (Quernheim and Knight, 2012b; Quernheim and Knight, 2012a; Chiang et al., 2013).

Applications. Just as syntactic parsing has found many unanticipated applications, we expect sembanks and statistical semantic processors to be used for many purposes. To get started, we are exploring the use of statistical NLU and NLG in

[5] UNL guidelines: www.undl.org/unlsys/unl/unl2005

a semantics-based machine translation (MT) system. In this system, we annotate bilingual Chinese/English data with AMR, then train components to map Chinese to AMR, and AMR to English. A prototype is described by Jones et al. (2012).

Disjunctive AMR. AMR aims to canonicalize multiple ways of saying the same thing. We plan to test how well we are doing by building AMRs on top of large, manually-constructed paraphrase networks from the HyTER project (Dreyer and Marcu, 2012). Rather than build individual AMRs for different paths through a network, we will construct highly-packed disjunctive AMRs. With this application in mind, we have developed a guideline[6] for disjunctive AMR. Here is an example:

```
(o / *OR*
   :op1 (t / talk-01)
   :op2 (m / meet-03)
   :OR (o2 / *OR*
          :mod (o3 / official)
          :arg1-of (s / sanction-01
                        :arg0 (s2 / state))))

official talks
state-sanctioned talks
meetings sanctioned by the state
```

AMR extensions. Finally, we would like to deepen the AMR language to include more relations (to replace :mod and :prep-X, for example), entity normalization (perhaps wikification), quantification, and temporal relations. Ultimately, we would like to also include a comprehensive set of more abstract frames like “Earthquake-01” (:magnitude, :epicenter, :casualties), “CriminalLawsuit-01” (:defendant, :crime, :jurisdiction), and “Pregnancy-01” (:father, :mother, :due-date). Projects like FrameNet (Baker et al., 1998) and CYC (Lenat, 1995) have long pursued such a set.

References

O. Abend and A. Rappoport. 2013. UCCA: A semantics-based grammatical annotation scheme. In *Proc. IWCS*.

C. Baker, C. Fillmore, and J. Lowe. 1998. The Berkeley FrameNet project. In *Proc. COLING*.

V. Basile, J. Bos, K. Evang, and N. Venhuizen. 2012a. Developing a large semantically annotated corpus. In *Proc. LREC*.

[6]Disjunctive AMR guideline: amr.isi.edu/damr.1.0.pdf

V. Basile, J. Bos, K. Evang, and N. Venhuizen. 2012b. A platform for collaborative semantic annotation. In *Proc. EACL demonstrations*.

A. Böhmová, J. Hajič, E. Hajičová, and B. Hladká. 2003. The Prague dependency treebank. In *Treebanks*. Springer.

A. Butler and K. Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology*, 7.

S. Cai and K. Knight. 2013. Smatch: An accuracy metric for abstract meaning representations. In *Proc. ACL*.

D. Chiang, J. Andreas, D. Bauer, K. M. Hermann, B. Jones, and K. Knight. 2013. Parsing graphs with hyperedge replacement grammars. In *Proc. ACL*.

D. Davidson. 1969. The individuation of events. In N. Rescher, editor, *Essays in Honor of Carl G. Hempel*. D. Reidel, Dordrecht.

M. Dreyer and D. Marcu. 2012. Hyter: Meaning-equivalent semantics for translation evaluation. In *Proc. NAACL*.

B. Jones, J. Andreas, D. Bauer, K. M. Hermann, and K. Knight. 2012. Semantics-based machine translation with hyperedge replacement grammars. In *Proc. COLING*.

H. Kamp, J. Van Genabith, and U. Reyle. 2011. Discourse representation theory. In *Handbook of philosophical logic*, pages 125–394. Springer.

P. Kingsbury and M. Palmer. 2002. From TreeBank to PropBank. In *Proc. LREC*.

D. B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11).

R. Martins. 2012. Le Petit Prince in UNL. In *Proc. LREC*.

C. M. I. M. Matthiessen and J. A. Bateman. 1991. *Text Generation and Systemic-Functional Linguistics*. Pinter, London.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1).

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, Philadelphia, PA.

S. Pradhan, E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2007. Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing (IJSC)*, 1(4).

D. Quernheim and K. Knight. 2012a. DAGGER: A toolkit for automata on directed acyclic graphs. In *Proc. FSMNLP*.

D. Quernheim and K. Knight. 2012b. Towards probabilistic acceptors and transducers for feature structures. In *Proc. SSST Workshop*.

K. Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

S. Shieber, F. C. N. Pereira, L. Karttunen, and M. Kay. 1986. Compilation of papers on unification-based grammar formalisms. Technical Report CSLI-86-48, Center for the Study of Language and Information, Stanford, California.

H. Uchida, M. Zhu, and T. Della Senta. 1996. UNL: Universal Networking Language—an electronic language for communication, understanding and collaboration. Technical report, IAS/UNU Tokyo.

H. Uchida, M. Zhu, and T. Della Senta. 1999. A gift for a millennium. Technical report, IAS/UNU Tokyo.

N. Venhuizen, V. Basile, K. Evang, and J. Bos. 2013. Gamification for word sense labeling. In *Proc. IWCS*.

R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. 2011. OntoNotes: A large training corpus for enhanced processing. In J. Olive, C. Christianson, and J. McCary, editors, *Handbook of Natural Language Processing and Machine Translation*. Springer.

Y. W. Wong and R. J. Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proc. HLT-NAACL*.