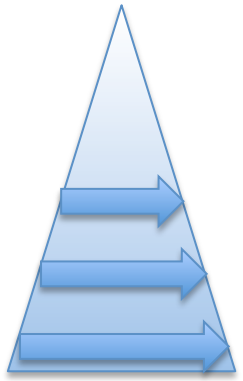
Semantics for SMT

Eduard Hovy
CMU

Historical significance of this workshop

- 1960s: direct replacement using all kinds of methods; Georgetown U and SYSTRAN
- 1970s: syntactic transfer; Eurotra
- 1980s: semantics and interlinguas; KBMT
- 1990s: SMT starts; Candide–Pangloss–LingStat
- 2000s: SMT exploration; Google Translate, lots of research prototypes, modern SYSTRAN
- 2010s: back to syntax…and semantics? — *wow*



Insofar as SMT produces acceptable translations, it is ALREADY handling semantics…

…it just doesn't know it is!

(there's nothing in principle that links source and translation, other than semantics)

It's all encoded in the translation lexicons/models and the sentence structure transformations

What we're asking for now is **explicit** representations of semantics … that's a whole different thing — do we really need it?

The Franz Och solution: Just continue with implicit encoding

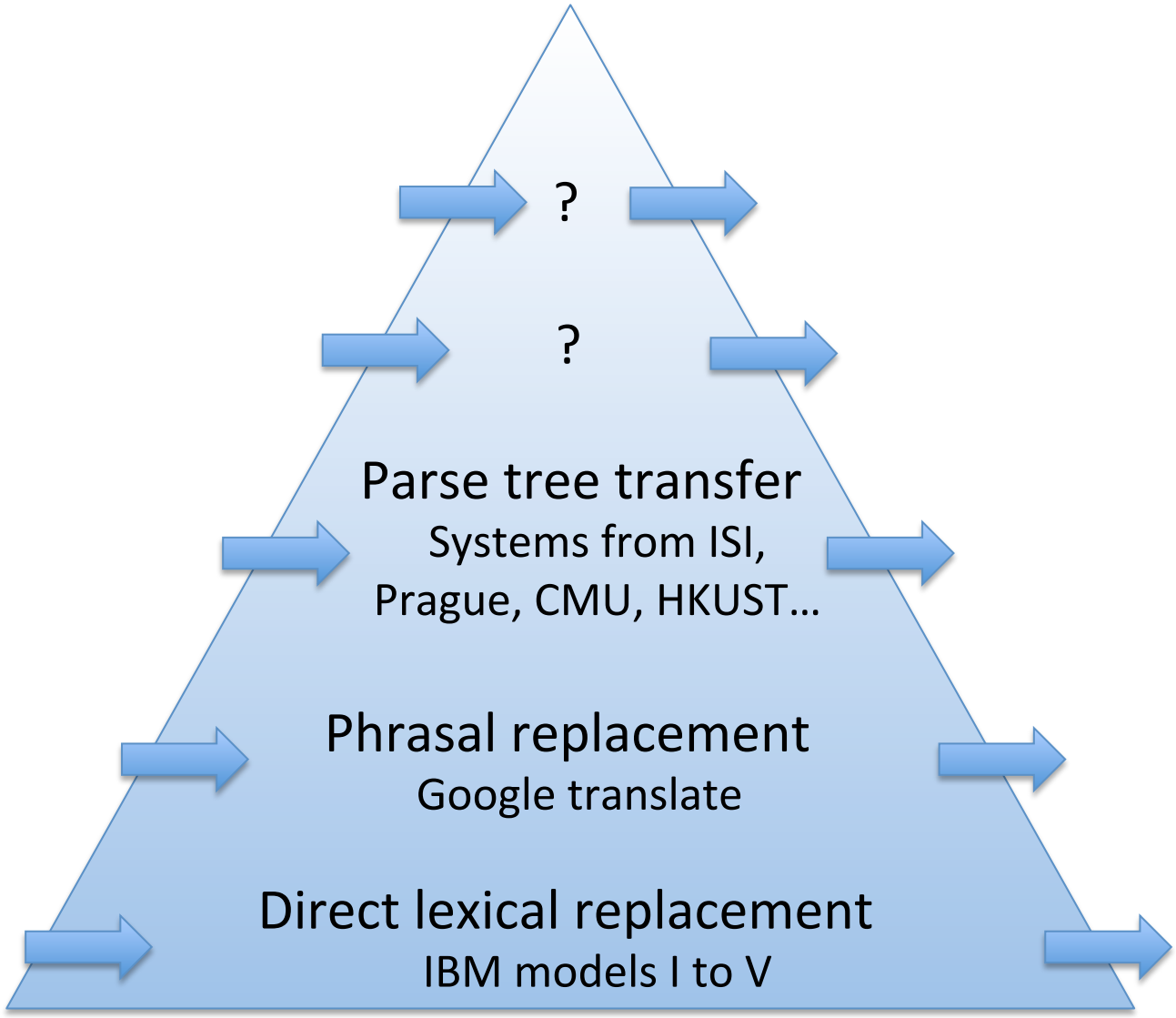
- Unigram translation table: bilingual dictionary
  - 200K words each side, = 2MB if each word is 10 chars (10 bytes)
- Bigram translation table (every bigram):
  - Lexicon: 200K words (not counting proper names)
  - Table entries: [200K × 200K words + translations] = 4×10^{10} entries
  - Each entry size = 4 words
  - 4×10^{10} entries × 40 bytes = 1.6×10^{12} bytes = 1.6×10^{3} TB
- Trigram translation table (every trigram):
  - 1.6×10^{8} TB
  - 2015: Backblaze storage 5c/GB ≈ $50/TB
- Better: store only attested ngrams (up to 5? 7? 9?), and fall back to shorter ones when not in table…
  - Carbonell et al. MT system…all 8grams of English

Why not do this?

- You end up with a **massive** table
- You have to deal with sparsity and novel constructions
- You still can't handle long-distance dependencies
- You still need special-purpose handling of [closed-class] phenomena like dates, numerical expressions, names, modality shifts, etc.

- Even just lemmatizing words and allowing a small amount of phrase breakup saves you tons of space….
- …but then you are on the way toward syntax … and semantics!

What kind(s) of semantics do you need to handle?



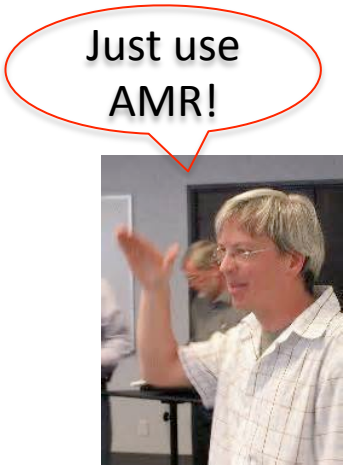
?…

?…

No reference;
deixis, quant…

No long-distance
dependencies,
dropped negation,
no reference…

The Kevin Knight solution: Use syntax, aim for more…



- Build dependency parse and add
  - argument roles (PropBank and more)
  - coref links
  - a little re-representation of modality
  - etc.
- SPL (AMR) was developed for English sentence generation, so it is still language-based (i.e., not yet close to deeper semantics) but at least gets rid of some pesky surface variations

Generally what semantics gives is the ability to rephrase when needed

- Deixis and reference
  - "they", "here", "this" —> expand to full description
  - "the soldiers shot the women and they fell/ they reloaded"
  - "first floor" —> "ground floor"
- Lexical paraphrase
  - "they crossed the river/road" —> "swam/ran across"
- Lexical gaps
  - Jp "連れション" —> Eng: "" (= "accompany to the toilet")
- Time
- Modality/certainty/factivity/etc.
  - "can/could/must" etc. — different social customs for this

More phenomena of semantics

Somewhat easier

Bracketing (scope) of predications

Word sense selection (incl. copula)

NP structure: genitives, modifiers…

Concepts: ontology definition

Concept structure (incl. frames and thematic roles)

Coreference (entities and events)

Pronoun classification (ref, bound, event, generic, other)

Identification of events

Temporal relations (incl. discourse and aspect)

Manner relations

Spatial relations

Direct quotation and reported speech

More difficult / 'deeper'

Quantifier phrases and numerical expressions

Comparatives

Coordination

Information structure (theme/rheme)

Focus

Discourse structure

Other adverbials (epistemic modals, evidentials)

Identification of propositions (modality)

Pragmatics/speech acts

Polarity/negation

Presuppositions

Metaphors

The necessary elements

- Semantic notation
- Semantic primitives
- Background knowledge instances
- Translation situation knowledge
- Inference

- Directed graph with nesting
- E.g, Framenet frames giving roles; ontology of relations and types
- E.g., freebase, Wikipedia…
- Instantiated graph combining text (discourse) and background info
- Rewriting transformations

Semantic substrate

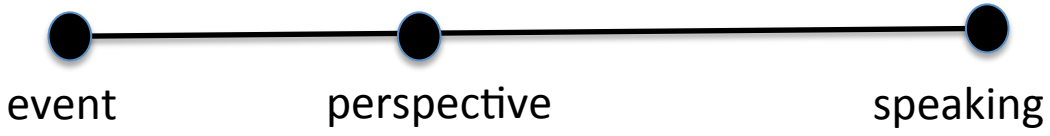
- **Notation**: Just some kind of dependency structure with cross-linking for coref
  - CCGs, HPSG's C-structure, universal dependencies, AMR, etc.
- **Rep primitives**:
  - **Entity** types: many ontologies or just Wordnet
  - **Event/state** types and frames: Framenet, etc.
  - **Qualities/attributes**: a bit of a problem, perhaps WN?
  - **Relations/links**: this is a problem (argN is not enough; but no good taxonomy exists)
  - **Special things** (negation, modality, time…): specialized solutions required

Knowledge

- Simple facts like in Freebase, PropStore, etc. to provide hints for reranking options:
  - People in China eat with chopsticks, so when you see "eat" and "chopsticks" it's likely to fill the :*instr* role
  - Pizza is eaten in Western countries mostly, so the :*instr* for an event whose :*patient* is pizza is unlikely to be chopsticks
- Useful types and their similarity groupings (ontologies or just term taxonomies like Wordnet) to handle sparsity, property inheritance, etc.

Special phenomena

- Time: Reichenbach's 3-point model
  - *event-time, speaking-time, perspective-time*
  - "by that time we had already left"



- Modality/certainty/factivity:
  - obligation (must), capability (can), certainty (might), etc.
- Focus/attention/theme/info structure:
  - "it was he who did the deed!"
  - "he did the deed"
  - "the deed was done by him"

Which one? Why?

The sad news

- **There is no single semantics**: there are many layers of increasingly deep semantics
- **There are many independent semantic phenomena**; each needs a separate solution

- Perhaps you can circumvent them [all] by some neat trick, like embeddings
- Otherwise you need to
  - Pick one or more phenomena
  - Design the representation
  - Analyze the problem and either build a special-purpose reasoner or define a model for learning
  - Fit in and evaluate