Semantics-based Machine Translation

Daniel Quernheim
daniel@ims.uni-stuttgart.de

Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

January 27, 2012

About this lecture

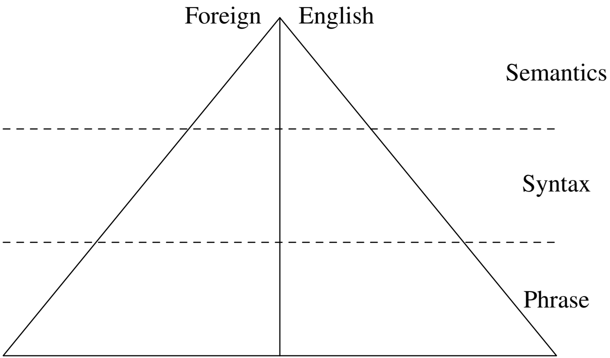
About me

- I am a Ph.D. student in Andreas Maletti's project
- Working on tree acceptors and transducers for syntax-based MT

About my ISI visit

- I visited USC/ISI for three months last year
- At ISI, I worked in KEVIN KNIGHT's group
- They produce state-of-the art results in syntax-based MT
- ...but they are working on *semantics-based MT* now!
- This lecture is mostly about **what they have in mind**, not what has happened already!

Motivation



Why semantics-based MT?

The more linguistic structure we use, the better the translation can be!

Motivation (2)

But what's wrong with phrase-based and syntax-based MT?

- We want to get the "**who did what to whom**" (WWW) right
- Preservation of meaning can be more important than grammaticality/fluency
- We are aiming for **useful** translation!

But haven't people tried and failed?

Yes, but…

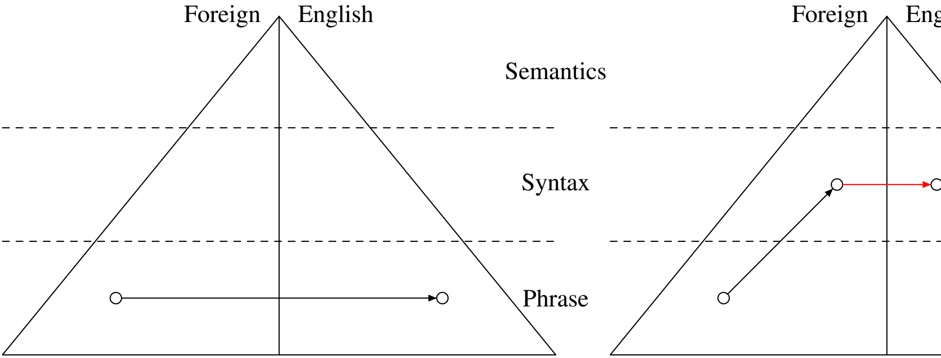
- that was before **statistics**
- small-scale, hand-crafted
- people said the same about syntax-based MT and look where it's now!

Words of wisdom

KEVIN KNIGHT: "As long as we get the **WWW** wrong, we are optimizing with respect to the wrong metric (BLEU)!"

WARREN WEAVER: "Thus it may be true that the way to translate from Chinese to Arabic [. . .] is not to attempt the direct route, shouting from tower to tower. Perhaps the way is to descend, from each language, down to the common base of human communication – the real but as yet undiscovered universal language – and then re-emerge by whatever particular route is convenient."

Different MT paradigms

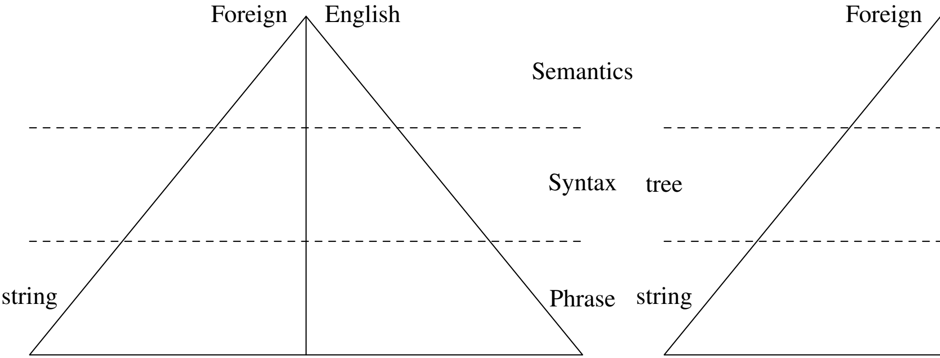


phrase-based MT: n-grammatical

syntax-based MT: grammatical

semantics-based MT: **sensible** and grammatical

Different MT paradigms (2)



Phrases: represented as **strings**

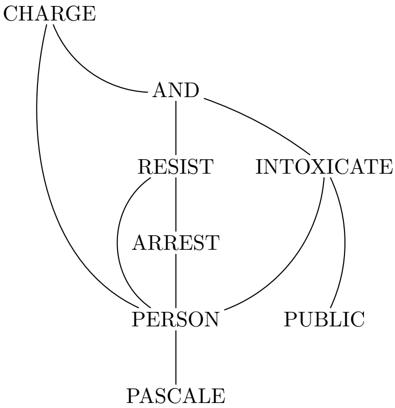
Syntax: represented by **trees**

Semantics: represented by **directed acyclic graphs**

Feature structures

$$
\begin{bmatrix}
\text{INSTANCE} & \text{charge} \\
\text{THEME} & \boxed{1} \begin{bmatrix} \text{INSTANCE} & \text{person} \\ \text{NAME} & \text{“Pascale”} \end{bmatrix} \\
\text{PRED} & \begin{bmatrix}
\text{INSTANCE} & \text{and} \\
\text{OP1} & \begin{bmatrix}
\text{INSTANCE} & \text{resist} \\
\text{AGENT} & \boxed{1} \\
\text{THEME} & \begin{bmatrix} \text{INSTANCE} & \text{arrest} \\ \text{THEME} & \boxed{1} \end{bmatrix}
\end{bmatrix} \\
\text{OP2} & \begin{bmatrix}
\text{INSTANCE} & \text{intoxicate} \\
\text{THEME} & \boxed{1} \\
\text{LOCATION} & \begin{bmatrix} \text{INSTANCE} & \text{public} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Directed acyclic graphs



$$
\begin{aligned}
\text{CHARGE} &\mapsto \text{charge}(\text{theme}, \text{pred}) \\
\text{AND} &\mapsto \text{and}(\text{op1}, \text{op2}) \\
\text{RESIST} &\mapsto \text{resist}(\text{agent}, \text{theme}) \\
\text{ARREST} &\mapsto \text{arrest}(\text{theme}) \\
\text{INTOXICATE} &\mapsto \text{intoxicate} \\
&\quad (\text{theme}, \text{location}) \\
\text{PUBLIC} &\mapsto \text{public}() \\
\text{PERSON} &\mapsto \text{person}(\text{name}) \\
\text{PASCALE} &\mapsto \text{"Pascale"}
\end{aligned}
$$

Translation pipelines

Syntax-based MT pipeline

$$\overset{\text{FSA}}{\text{fstring}} \rightarrow \overset{\text{nl-XTOPs}^{-1}}{\boxed{\text{translate}}} \rightarrow \overset{\text{RTG}}{\text{etree}} \rightarrow \overset{\text{FSA}}{\boxed{\text{language model}}} \rightarrow \overset{\text{FSA}}{\text{estring}}$$

- The individual components are efficiently represented as **weighted tree acceptors and transducers.**

$$\begin{aligned}\text{estring} = \textsc{BestPath}(\textsc{Intersect}(&\text{language model},\\ &\textsc{Yield}(\textsc{Backwards}(\text{translate}, \text{fstring})))).\end{aligned}$$

Translation pipelines (2)

Semantics-based MT pipeline

$$\text{fstring} \rightarrow \boxed{\text{understand}} \rightarrow \text{esem} \rightarrow \boxed{\text{rank}} \rightarrow \text{esem}$$
$$\rightarrow \boxed{\text{generate}} \rightarrow \text{etree} \rightarrow \boxed{\text{rank}} \rightarrow \text{estring}$$

- No suitable automaton framework is known!

Algorithms and automata

	string automata	tree automata	graph automata
k-best	paths through a WFSA	trees in a weighted forest	**?**
EM training	Forward-backward EM	Tree transducer EM training	**?**
Determinization	of weighted string acceptors	of weighted tree acceptors	**?**
Transducer composition	WFST composition	Many transducers not closed under composition	**?**
General tools	AT&T FSM, Carmel, OpenFST	Tiburon	**?**

Table: General-purpose algorithms for strings, trees and feature structures.

Algorithms and automata (2)

Our goal

- Find an adequate automaton model for the pipeline parts
- Investigate algorithms and fill all the blanks!

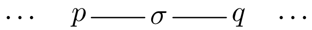
Candidates

- Treating everything as a tree (too weak?)
- Unification grammars (HPSG, LFG) (too powerful?)
- Hyperedge replacement grammar (too powerful?)
- Some straightforward extension of tree automata?

Dag automata

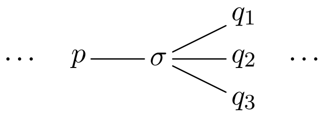
finite string automaton: (FSA)

one input state, one input symbol, one output state



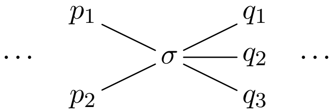
finite tree automaton: (FTA)

one input state, one input symbol, many output states



finite dag automaton: (FDA?)

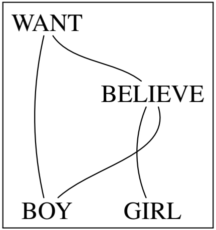
many input states, one input symbol, many output states



Dag automata (2)

Kamimura and Slutzki (1981, 1982)

- Dag acceptors and dag-to-tree transducers
- They proved a couple of technical properties, no algorithms
- We investigate their model with some adjustments:
  - not only adjacent leaves can be connected
  - top-down transducers instead of bottom-up
  - we add weights (probabilities)

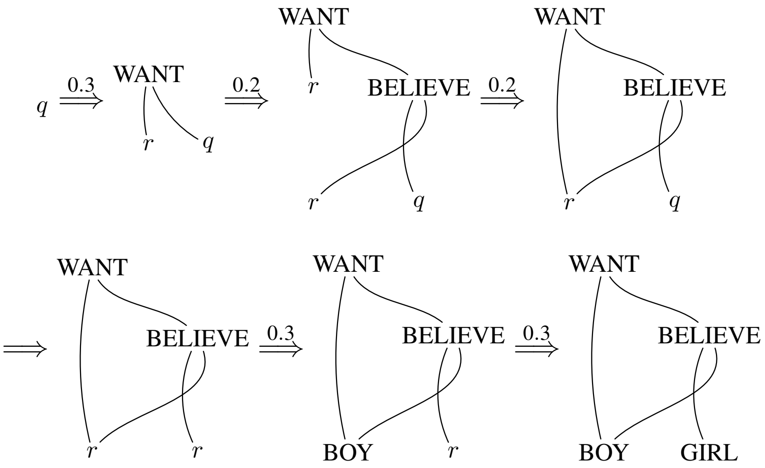


Example dag automaton

$$
\begin{aligned}
q &\rightarrow \mathsf{WANT}(r, q)\langle 0.3\rangle \\
q &\rightarrow \mathsf{BELIEVE}(r, q)\langle 0.2\rangle \\
q &\rightarrow r\,\langle 0.4\rangle \mid \emptyset\,\langle 0.1\rangle \\
r &\rightarrow \mathsf{BOY}\langle 0.3\rangle \mid \mathsf{GIRL} \\
&\quad\ \langle 0.3\rangle \mid \emptyset\,\langle 0.1\rangle \\
[r, r] &\rightarrow r\,\langle 0.2\rangle \\
[r, r, r] &\rightarrow r\,\langle 0.1\rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{WANT} &\mapsto \text{want}(\text{agent}, \text{theme}) \\
\mathsf{BELIEVE} &\mapsto \text{believe}(\text{agent}, \text{theme}) \\
\mathsf{BOY} &\mapsto \text{boy}() \\
\mathsf{GIRL} &\mapsto \text{girl}()
\end{aligned}
$$

Example dag generation



Example dag transducer rules

- Rules have m incoming edges with states and produce m trees
- Rules have n outgoing edges and n variables to pass states down

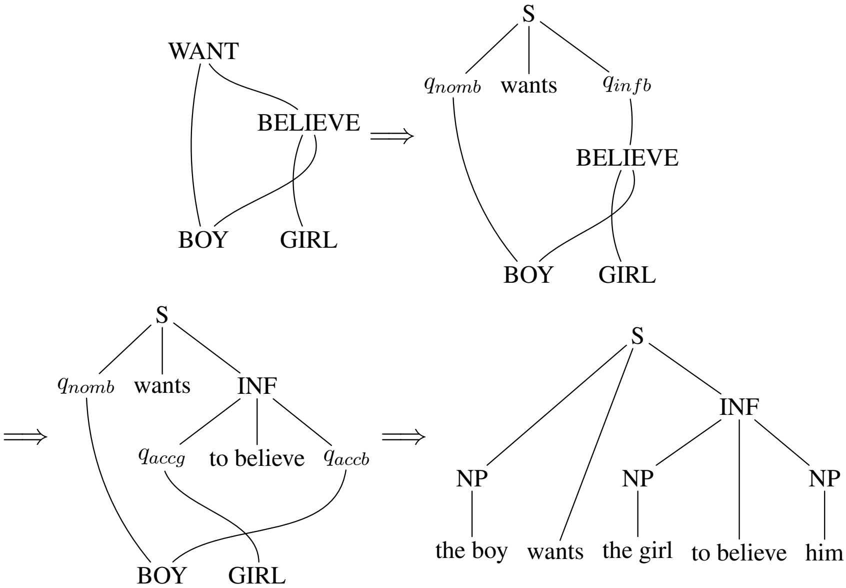
$$[q_{nomb}, q_{accb}].\text{BOY} \rightarrow \text{NP(the boy)}, \text{NP(him)}$$

$$q_{accg}.\text{GIRL} \rightarrow \text{NP(the girl)}$$

$$q_s.\text{WANT}(x, y) \rightarrow \text{S}(q_{nomb}.x, \text{wants}, q_{infb}.y)$$

$$q_{infb}.\text{BELIEVE}(x, y) \rightarrow \text{INF}(q_{accg}.x, \text{to believe}, q_{accb}.y)$$

Example dag transduction



Toolkit

I implemented in Python...

- unweighted and weighted **membership checking**
- unweighted and weighted **dag-to-tree transductions**

- packing the **set of derivations** into a dag acceptor
- packing the **set of output trees** into an RTG

- unweighted and weighted ***n*-best generation**

- **backward application** (tree to dag)

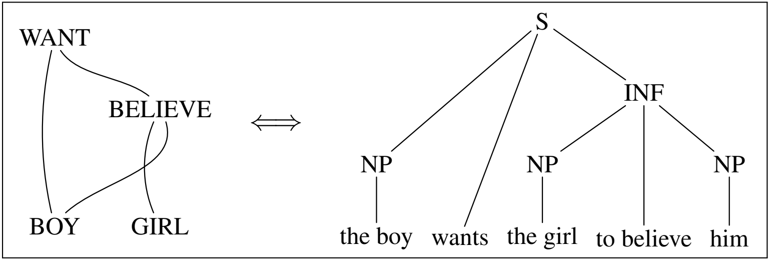
- product construction: **intersection and union**

- nice **visualization** of trees and graphs using GraphViz

Building an NLP system

With the theoretical background, it should be possible to carry out the same program that worked for syntax-based MT:

- Collect lots of training data



- Train models for parts of the translation pipeline
- Use them in a bucket-brigade approach or in an integrated decoder

Where does our training data come from?

Training data

- Goal: gold standard esem bank
- In the meantime: annotate data automatically using other resources (e.g. Propbank/OntoNotes) and manually correct them

After training: evaluation

Is BLEU the right metric?

BLEU and other n-gram based automated metrics...

- ...favor translations that make the same lexical choices as the reference translations
- ...capture translation fluency, but often disagree with human judgment
- ...are still the metrics of choice of most people!

What makes a good metric

- It should favor useful (meaning-preserving) translations
- It should not require identical lexical choices
- It should be relatively cheap

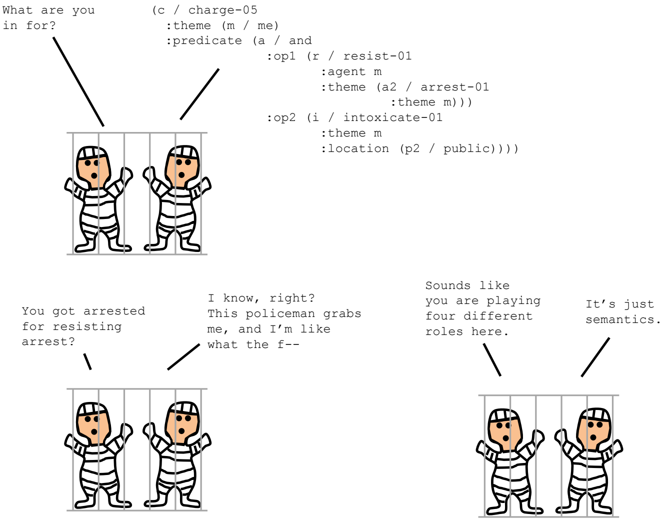
A semantically motivated metric

MEANT (Lo and Wu 2011)

- measures accuracy (precision and recall) of semantic frames
- → it scores the **who did what to whom**
- can be performed by monolinguals, no bilinguals needed
- less labor-intensive than other adequacy-oriented metrics
- good correlation coefficient with human judgment

The ~~end~~ beginning

Thank you for your attention! – Questions?



©KEVIN KNIGHT