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

INSTANCE charge

THEME

INSTANCE person

NAME "Pascale"

INSTANCE and

OP1 INSTANCE resist

AGENT 1

THEME

OP2 INSTANCE intoxicate

THEME 1

LOCATION

INSTANCE public

INSTANCE arrest

THEME 1
Directed acyclic graphs

CHARGE $\mapsto$ charge(theme, pred)
AND $\mapsto$ and(op1, op2)
RESIST $\mapsto$ resist(agent, theme)
ARREST $\mapsto$ arrest(theme)
INTOXICATE $\mapsto$ intoxicate((theme, location))
PUBLIC $\mapsto$ public()
PERSON $\mapsto$ person(name)
PASCALE $\mapsto$ "Pascale"
Translation pipelines

Syntax-based MT pipeline

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

\[ \text{estring} = \text{BESTPATH}(\text{INTERSECT}(\text{language model}, \text{YIELD}(\text{BACKWARDS}(\text{translate}, \text{fstring}))))). \]
Translation pipelines (2)

Semantics-based MT pipeline

\[
\text{fstring} \rightarrow \text{understand} \rightarrow \text{esem} \rightarrow \text{rank} \rightarrow \text{esem} \\
\rightarrow \text{generate} \rightarrow \text{etree} \rightarrow \text{rank} \rightarrow \text{estring}
\]

- No suitable automaton framework is known!
## Algorithms and automata

<table>
<thead>
<tr>
<th></th>
<th>string automata</th>
<th>tree automata</th>
<th>graph automata</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-best</td>
<td>paths through a WFSA</td>
<td>trees in a weighted forest</td>
<td></td>
</tr>
<tr>
<td>EM training</td>
<td>Forward-backward EM</td>
<td>Tree transducer EM training</td>
<td></td>
</tr>
<tr>
<td>Determinization</td>
<td>of weighted string acceptors</td>
<td>of weighted tree acceptors</td>
<td></td>
</tr>
<tr>
<td>Transducer composition</td>
<td>WFST composition</td>
<td>Many transducers not closed under composition</td>
<td></td>
</tr>
<tr>
<td>General tools</td>
<td>AT&amp;T FSM, Carmel, OpenFST</td>
<td>Tiburon</td>
<td></td>
</tr>
</tbody>
</table>

**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
  \[ \cdots \quad p \xrightarrow{\sigma} q \quad \cdots \]

finite tree automaton: (FTA)
  one input state, one input symbol, many output states
  \[ \cdots \quad p \xrightarrow{\sigma} q_1 \quad q_2 \quad \cdots \]

finite dag automaton: (FDA?)
  many input states, one input symbol, many output states
  \[ \cdots \quad p_1 \xrightarrow{\sigma} q_1 \quad q_2 \quad \cdots \quad p_2 \xrightarrow{\sigma} q_2 \quad q_3 \quad \cdots \]
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

\[
\begin{align*}
q & \rightarrow \text{WANT}(r, q) \langle 0.3 \rangle \\
q & \rightarrow \text{BELIEVE}(r, q) \langle 0.2 \rangle \\
q & \rightarrow r \langle 0.4 \rangle \ | \ \emptyset \langle 0.1 \rangle \\
r & \rightarrow \text{BOY} \langle 0.3 \rangle \ | \ \text{GIRL} \langle 0.3 \rangle \ | \ \emptyset \langle 0.1 \rangle \\
[r, r] & \rightarrow r \langle 0.2 \rangle \\
[r, r, r] & \rightarrow r \langle 0.1 \rangle \\
\end{align*}
\]

\[
\begin{align*}
\text{WANT} & \mapsto \text{want}(\text{agent, theme}) \\
\text{BELIEVE} & \mapsto \text{believe}(\text{agent, theme}) \\
\text{BOY} & \mapsto \text{boy()} \\
\text{GIRL} & \mapsto \text{girl()} \\
\end{align*}
\]
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

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

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

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

\[
q_{\text{infb}}.\text{BELIEVE}(x, y) \rightarrow \text{INF}(q_{\text{accg}}.x, \text{to believe}, q_{\text{accb}}.y)
\]
Example dag transduction

\[
\begin{align*}
S & \quad \Rightarrow \\
q_{nomb} & \quad \text{wants} \\
q_{inf} & \\
\text{BELIEVE} & \\
\text{BOY} & \quad \text{GIRL} \\
S & \quad \Rightarrow \\
q_{nomb} & \quad \text{wants} \\
q_{inf} & \\
\text{BELIEVE} & \\
\text{BOY} & \quad \text{GIRL} \\
\Rightarrow \\
S & \quad \Rightarrow \\
q_{nomb} & \quad \text{wants} \\
q_{accg} & \quad \text{to believe} \\
q_{acb} & \\
\text{BOY} & \quad \text{GIRL} \\
\Rightarrow \\
\text{NP} & \quad \text{NP} & \quad \text{NP} \\
\text{the boy} & \quad \text{wants} & \quad \text{the girl} & \quad \text{to believe} & \quad \text{him} \\
\end{align*}
\]
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

Diagram:

```
S
  ▼
  NP  NP  NP
  ▼  ▼  ▼
  the boy wants the girl to believe him

WANT
  ▼
  BELIEVE
  ▼
  BOY  GIRL
```

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?

What are you in for?

(c / charge-05
 :theme (m / me)
 :predicate (a / and
 :op1 (r / resist-01
 :agent m
 :theme (a2 / arrest-01
 :theme m)))
 :op2 (i / intoxicate-01
 :theme m
 :location (p2 / public))))

You got arrested for resisting arrest?

I know, right? This policeman grabs me, and I’m like what the f--

Sounds like you are playing four different roles here.

It’s just semantics.