

Syntax-Based Translation: Good News, Bad News, How to Win Big

Adam Lopez

with thanks to Ondřej Bojar
(and apologies to Richard P. Gabriel)

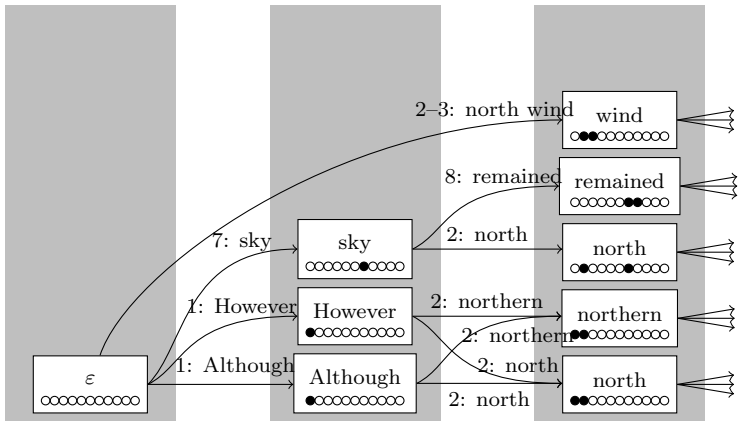
- ▶ Why do we care about syntax-based MT?
- ▶ How does it work?
- ▶ What are the open problems?

- ▶ Why do we care about syntax-based MT?
- ▶ How does it work?
- ▶ What are the open problems?

Disclaimer

Fast-moving field, we only scratch the surface

虽然 北 风 呼啸 ， 但 天空 依然 十分 清澈 。
Although north wind howls , but sky still extremely limpid .
 1 2 3 4 5 6 7 8 9 10 11

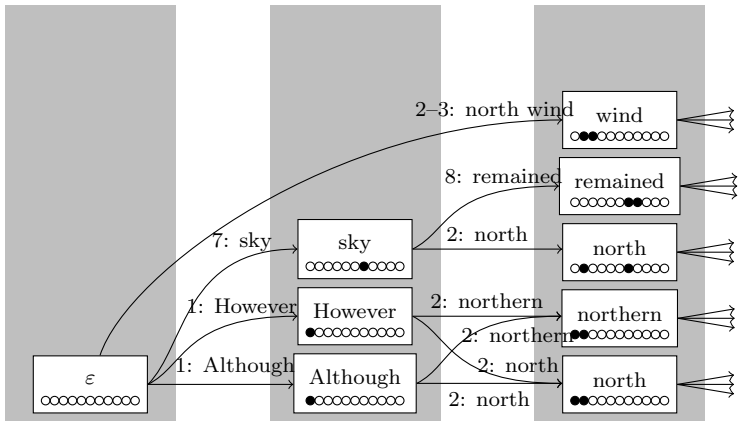


However , the sky remained clear under the strong north wind .

虽然 北 风 呼 啸 ， 但 天 空 依 然 十 分 清 澈 。

Although north wind howls , but sky still extremely limpid .

虽然 北 风 呼啸 ， 但 天空 依然 十分 清澈 。
Although north wind howls , but sky still extremely limpid .
 1 2 3 4 5 6 7 8 9 10 11



Phrase-based models are good, but not perfect

- ▶ computing all possible reorderings is NP-complete
- ▶ can't generalize
- ▶ can't model long-distance dependencies
- ▶ can't model grammaticality

Good News

Syntax-based models aim to solve these problems

- ▶ polynomial complexity
- ▶ can generalize
- ▶ can model long-distance dependencies
- ▶ can model grammaticality

the



la

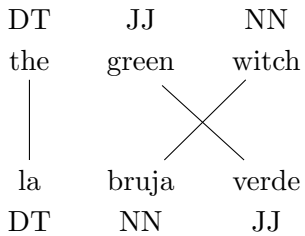
green

witch



bruja

verde



DT JJ NN

|

X

DT NN JJ

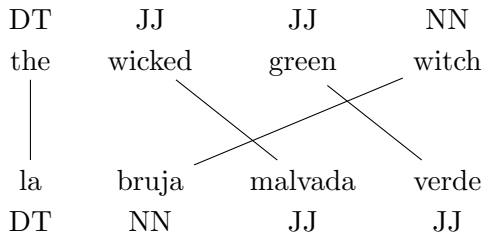
DT JJ NN

|

X

DT NN JJ

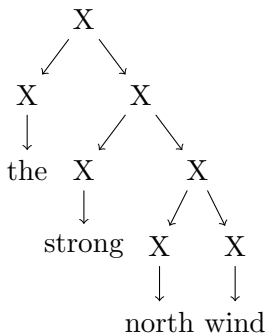
NP \longrightarrow DT₁JJ₂NN₃/DT₁NN₃JJ₁

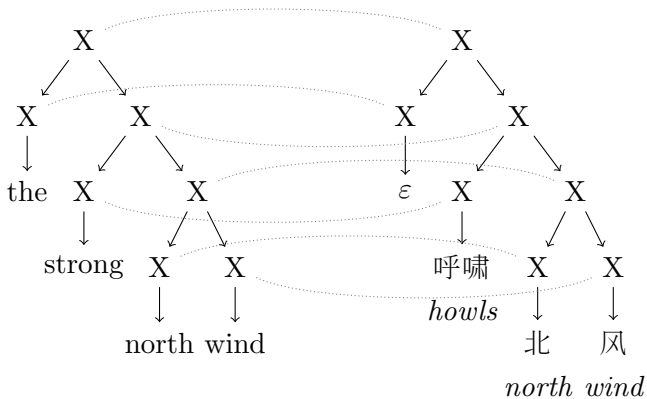


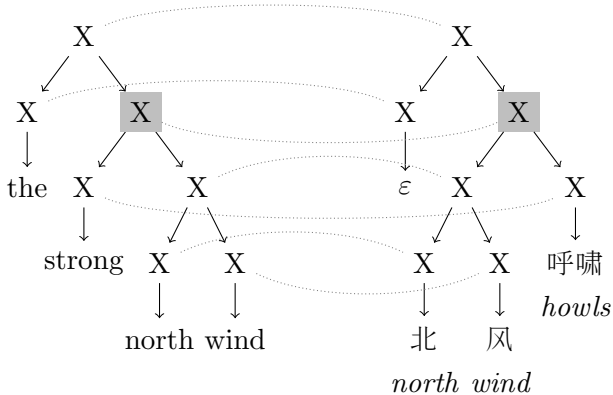
NP \longrightarrow DT₁JJ₂JJ₃NN₄/DT₁NN₄JJ₂JJ₃

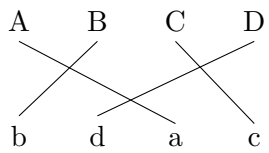
Problem Phrase-based decoding with full reordering has exponential complexity.

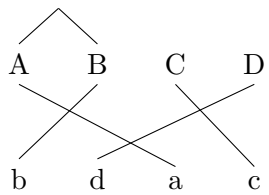
Idea Use binary-bracketing SCFG for polynomial complexity.

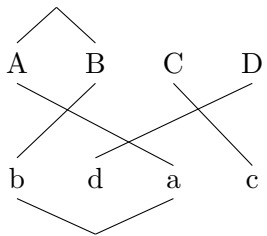


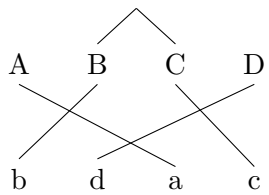


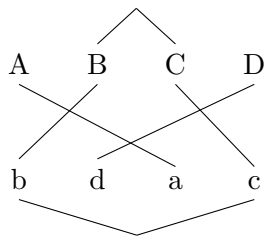


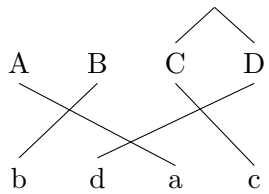


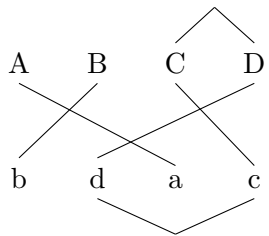


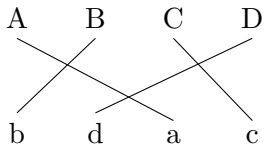




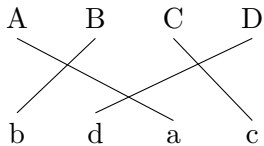




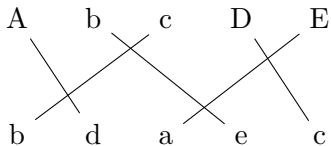




not possible with binary SCFG



not possible with binary SCFG



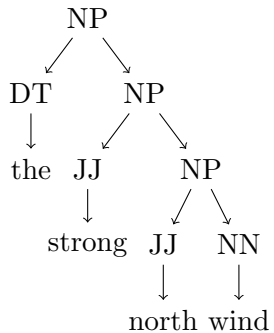
not possible with 4-ary SCFG

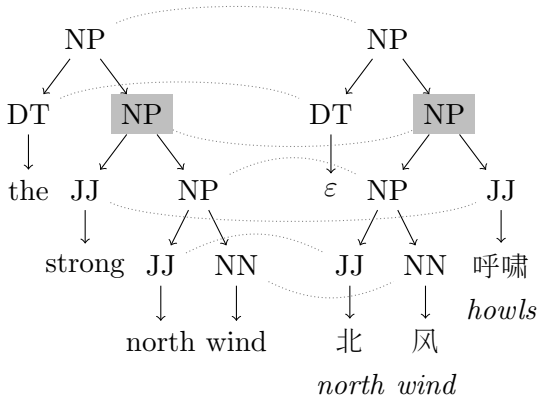
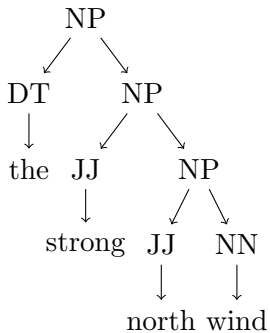
Syntax-based models aim to solve these problems

- ▶ polynomial complexity
- ▶ can generalize
- ▶ can model long-distance dependencies
- ▶ can model grammaticality

Problem Phrase-based cannot model grammaticality.

Idea Constrain SCFG to target-side syntax.





Problem Stack decoding doesn't apply

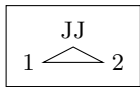
Problem Stack decoding doesn't apply

Idea Decoding is parsing

北
north (2)

风
wind (3)

呼啸
howls (4)



JJ → 北 / north

北

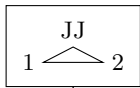
north (2)

风

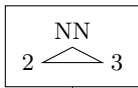
wind (3)

呼啸

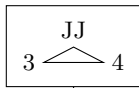
howls (4)



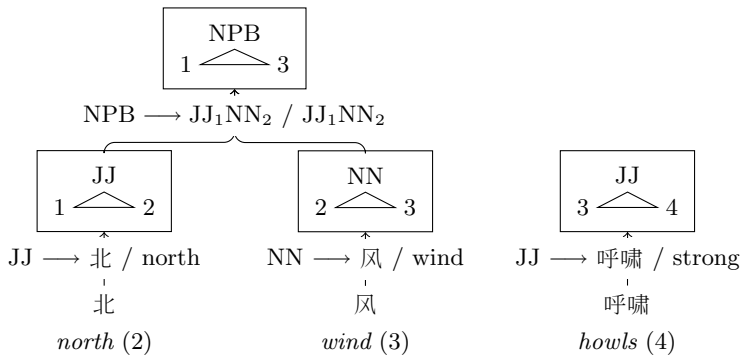
JJ → 北 / north
|
北
north (2)

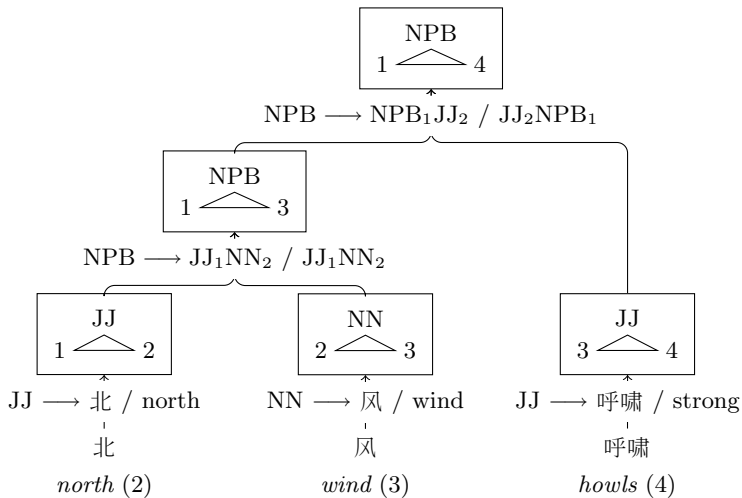


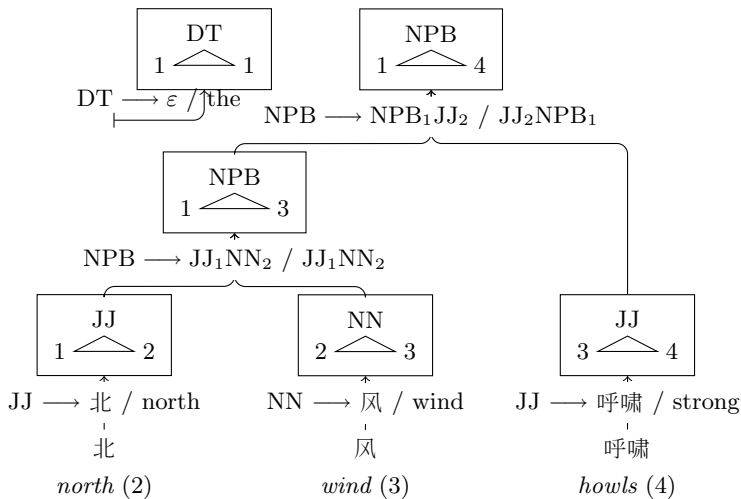
NN → 风 / wind
|
风
wind (3)

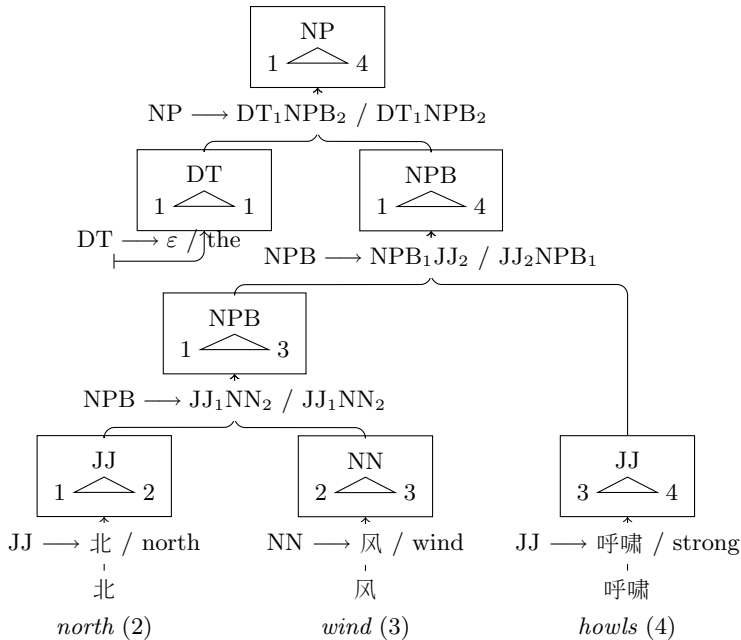


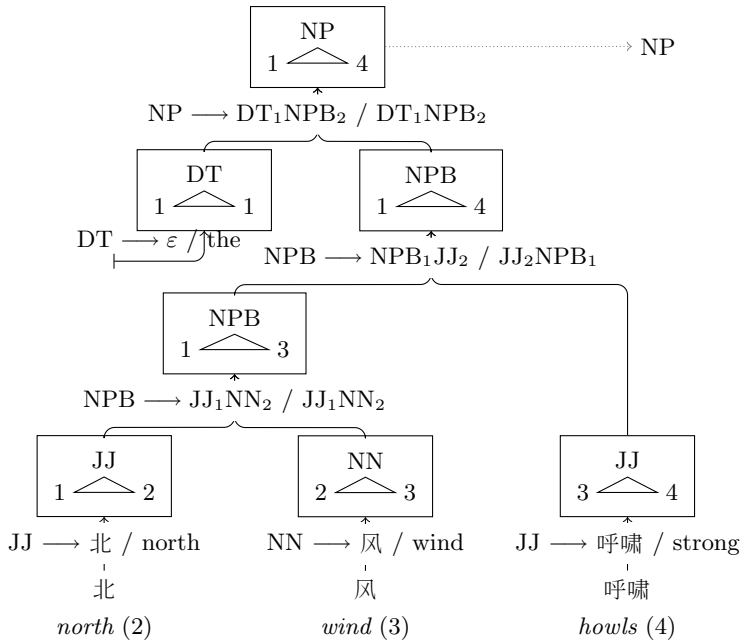
JJ → 呼啸 / strong
|
呼啸
howls (4)

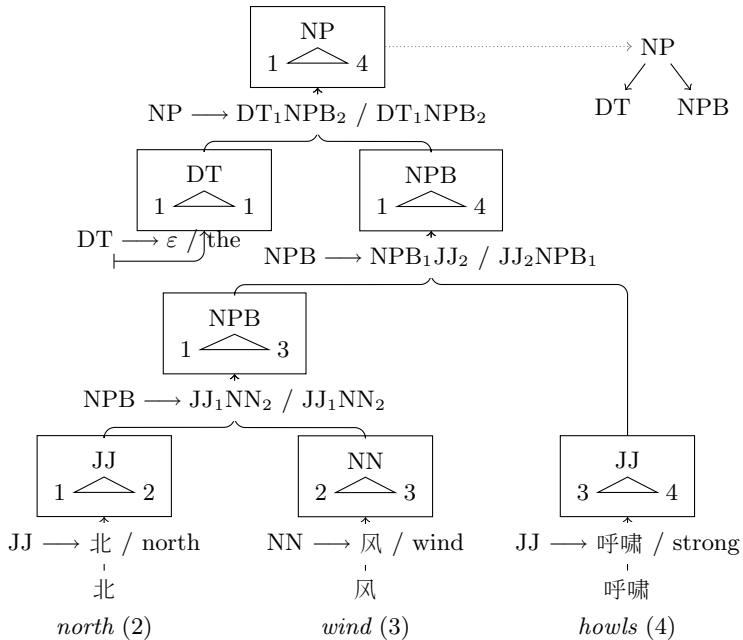


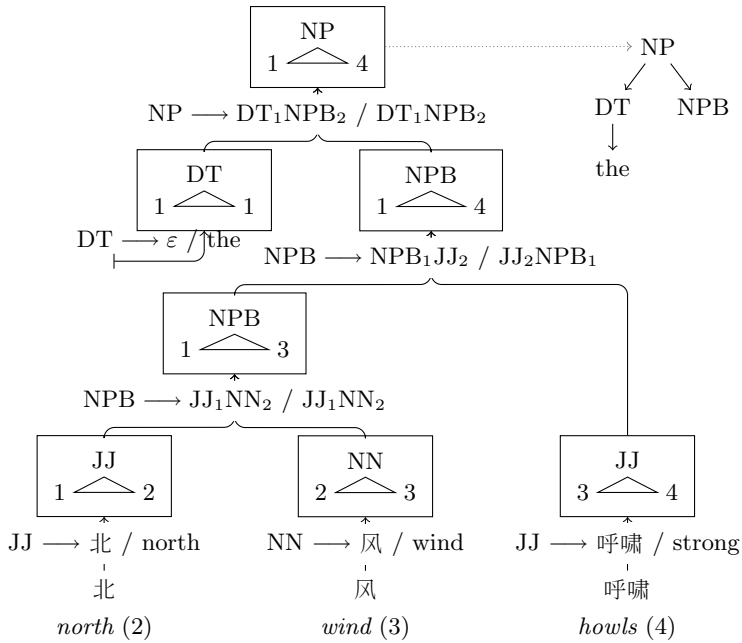


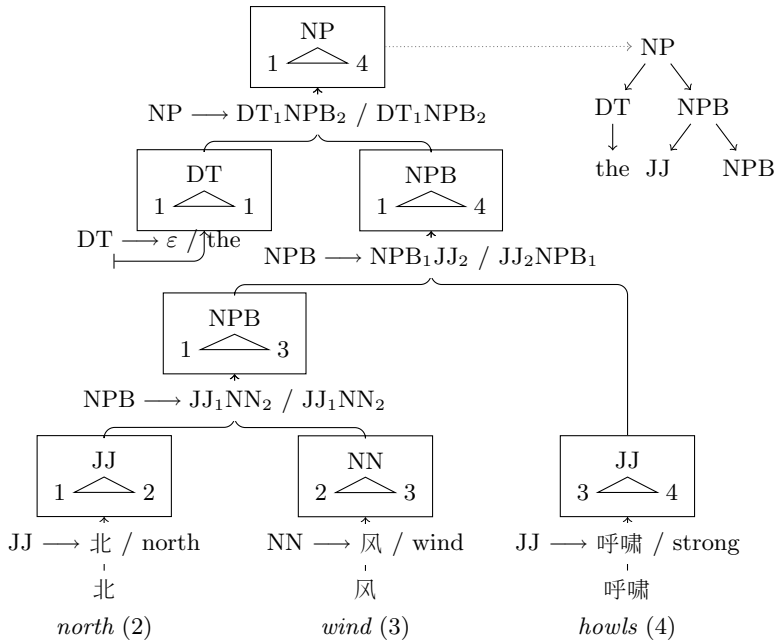


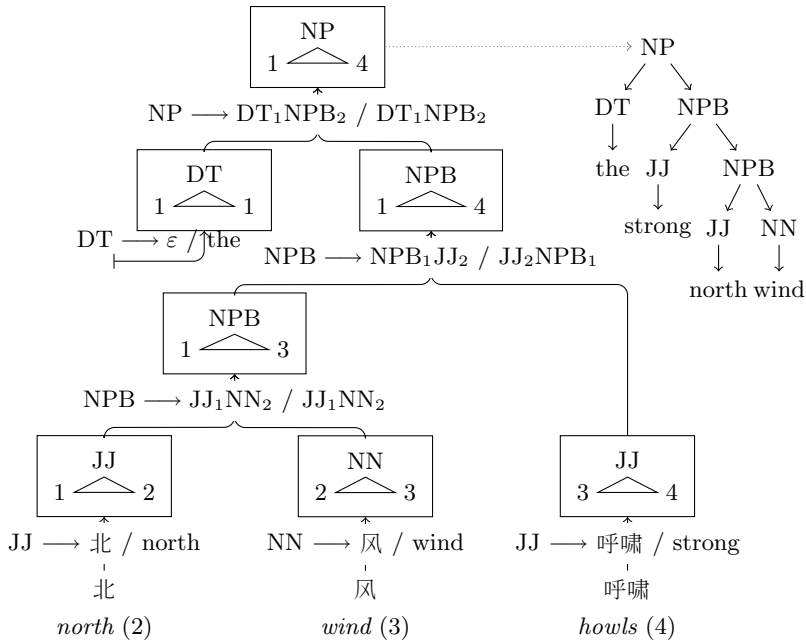












Bad News

It doesn't really work.

- ▶ Bracketing grammar doesn't capture all alignments.
- ▶ Tree isomorphism at production level too strict for real syntax.
- ▶ Language models are still important.
- ▶ Much higher cost in hierarchical vs. phrase-based.

It doesn't really work.

- ▶ Bracketing grammar doesn't capture all alignments.
- ▶ Tree isomorphism at production level too strict for real syntax.
- ▶ Language models are still important.
- ▶ Much higher cost in hierarchical vs. phrase-based.

Where do we go next?

- ▶ More theory?
- ▶ More articulated models?

Modeling translational equivalence using weighted finite state transducers is like approximating a high-order polynomial with line segments... the relatively low expressive power of weighted finite state transducers limits the quality of SMT systems.

–Burbank *et al.* 2005

But language is hierarchical.

–anonymous MT researcher

I think phrases are a passing fad.

–anonymous MT researcher

This type of difficulty has happened in other research areas.

See: “Lisp: Good News, Bad News, How to Win Big”, presented at the EUROPAL conference by Richard P. Gabriel in 1989.

This type of difficulty has happened in other research areas.

See: “Lisp: Good News, Bad News, How to Win Big”, presented at the EUROPAL conference by Richard P. Gabriel in 1989.

Lisp = syntax-based models

Unix and C++ = phrase-based models

- Simplicity** the design must be simple, both in implementation and interface. It is more important for the interface to be simple than the implementation.
- Correctness** the design must be correct in all observable aspects. Incorrectness is simply not allowed.
- Consistency** the design must not be inconsistent. A design is allowed to be slightly less simple and less complete to avoid inconsistency. Consistency is as important as correctness.
- Completeness** the design must cover as many important situations as is practical. All reasonably expected cases must be covered. Simplicity is not allowed to overly reduce completeness.

Simplicity the design must be simple, both in implementation and interface. It is more important for the interface to be simple than the implementation.

Correctness

The Right Thing

Consistency

allowed to be slightly less simple and less complete to avoid inconsistency. Consistency is as important as correctness.

Completeness the design must cover as many important situations as is practical. All reasonably expected cases must be covered. Simplicity is not allowed to overly reduce completeness.

- Simplicity** the design must be simple. Simplicity is the most important consideration in a design.
- Correctness** the design must be correct in all observable aspects. It is slightly better to be simple than correct.
- Consistency** the design must not be overly inconsistent. It is better to drop those parts of the design that deal with less common circumstances than to introduce either implementational complexity or inconsistency.
- Completeness** the design must cover as many important situations as is practical. Completeness can be sacrificed in favor of any other quality. In fact, completeness must be sacrificed whenever implementation simplicity is jeopardized.

Simplicity the design must be simple. Simplicity is the most important consideration in a design.

Correctness the design must be correct in all observable aspects. It is slightly better to be simple than correct.

Consistency the design must be consistent with the requirements, either implementational complexity or inconsistency.

Completeness the design must cover as many important situations as is practical. Completeness can be sacrificed in favor of any other quality. In fact, completeness must be sacrificed whenever implementation simplicity is jeopardized.

Worse is Better

The good news is that in 1995 we will have a good operating system and programming language. The bad news is that they will be Unix and C++.

–Richard Gabriel

The good news is that in 1995 we will have a good operating system and programming language. The bad news is that they will be Unix and C++.

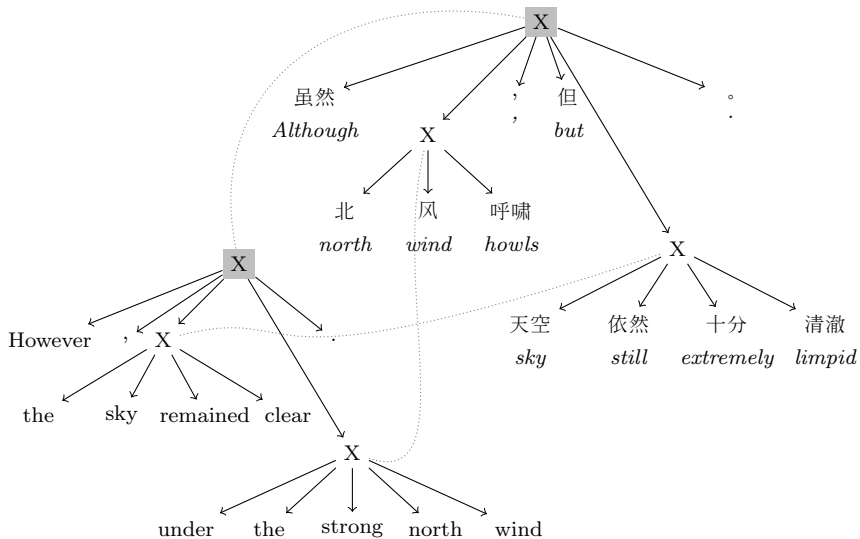
–Richard Gabriel

In 2018, will we have a good translation system based on phrases?

How to Win Big

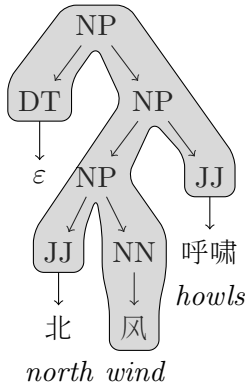
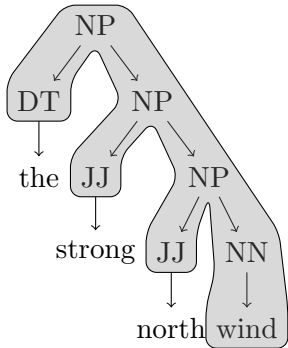
Observation Phrase-based models good at local reordering.

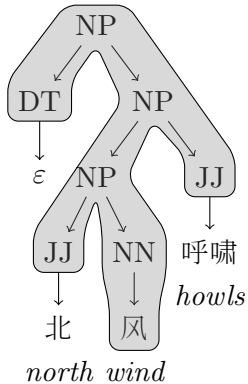
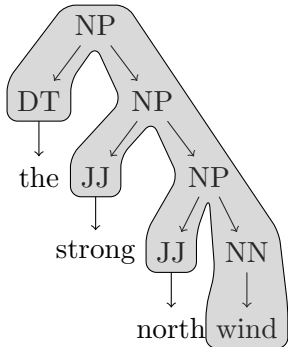
Idea Use phrases to reorder phrases.



Observation Phrase-based models good, but not grammatical.

Idea Add syntax, but keep the phrases.





$X_{245} \rightarrow DT_1 JJ_1 JJ_2 wind / DT_1 JJ_2 风 JJ_1$

Current status

- ▶ Syntax-based models competitive with phrase-based
 - ▶ Slightly better for Chinese-English
 - ▶ Slightly worse for Arabic-English
 - ▶ Open question for European languages
 - ▶ Language models make a bigger difference
- ▶ Not as fast as advertised
 - ▶ With 5-gram language model – $O(n^{11})$
 - ▶ Easy tricks in phrase-based models not applicable
 - ▶ Work on clever search algorithms
- ▶ Parsing progress – 1997: 88.1%, 2007: 92.4%

Many, many more angles

- ▶ Different formal models with different properties
 - ▶ Dependency grammar
 - ▶ Synchronous tree substitution grammar
 - ▶ Synchronous tree adjoining grammar
- ▶ Parsing: source, target, or both?
- ▶ How do we train a model?

See handout for some further reading

Take home messages

- ▶ Phrase-based models may not solve all of our problems
- ▶ Syntax-based models attempt to solve many of these problems directly
 - ▶ But practice lags behind theory
- ▶ Formal models from parsing = translation by parsing
- ▶ Lots of interesting open problems

Acknowledgements:

Necip Fazil Ayan, Phil Blunsom, Ondřej Bojar, David Chiang,
Trevor Cohn, Bonnie Dorr, Chris Dyer, Philipp Koehn, Nitin
Madnani, Christof Monz, Miles Osborne, Philip Resnik, and
Michael Subotin

Any errors are my own.