

TectoMT for Plaintext Freaks



Ondřej Bojar

bojar@ufal.mff.cuni.cz

Institute of Formal and Applied Linguistics

Faculty of Mathematics and Physics

Charles University, Prague

- Motivation: Large-scale rich NLP.
- Achievements: CzEng and Czech monolingual corpus parsed.
- HowTo: Which bits of TectoMT you need.
 - Caveats: Mind your NFS.
- Debugging someone else's code.
- Applications: Suggestions for the MT Marathon week.

TectoMT is great:

- Bindings to many tools (taggers, parsers, aligners, . . .).
- Bindings *between* the tools.
- Easy to build pipelines.
- Easy to hack at various layers of NLP.

TectoMT was horrible:

- Rather verbose XML file format.
- Rather funny startup: init environment, then bash aliases to launch “Perl wrapped in btred” \Rightarrow pain to parallelize.
- Inevitable to debug someone else’s code!

Achievements



Sun Grid Engine on 40 4-CPU computers.

We were able to annotate big Czech monolingual corpus:

Total sentences	51.6 mil.
Sentences with a t-tree	51.1 mil.
<hr/>	
a-nodes, i.e. tokens	0.86 mld. (Gword)
t-nodes	0.60 mld. (G)
<hr/>	
files	> 1 mil.
disk space in tree format (.tmt.gz)	72GB
disk space in tab-delimited rich export (.txt.gz)	17GB

Data sources: Czech National Corpus 73%, Web Collection 17%,
WMT09 Monolingual Training Data 10%

We also parsed and aligned CzEng (Bojar et al., 2008a), an extended version of 7 million Czech-English parallel sentences.

HowTo: Plaintext to TMT



TectoMT's file format is called TMT:

- XML, an application of PML (Pajas and Štěpánek, 2005).

⇒ The first step needed is to wrap plaintext with XML tags.

...

```
<LM id='news-dev2009a-00-s8'>
```

```
  <english_source_sentence> Government crisis coming , says Gallup...
```

```
  <czech_source_sentence> Gallup vidí vládní krizi</czech_source_s...
```

```
</LM>
```

...

E.g. `tools/format_convertors/czeng07_to_tmt/czeng07_to_tmt.pl`.

- Avoid > 50 to 100 sentences in a file.
- Avoid > 1000 files in a directory.

⇒ Clever convertors create nested directory structure.

HowTo: Scenarios on Grid



1. Create filelist: `find dir -name '*.tmt.gz' > filelist`
2. Submit parallel execution of a TectoMT scenario:

```
tools/cluster_utils/qrunblocks \  
  filelist \  
  "Miscel::SuicideIfMemFull Miscel::SuicideIfDiskFull Block1 Block2 ..." \  
  --jobs 20 --attempts 200 \  
  --finished-contains "SCzechT"
```

- Suicides protect your environment.
- `--attempts` restart your jobs after suicides or random deaths.
- `--finished-contains` skips files that seem to contain the desired bit.
- Jobs run independently in the background.
- Independent log files (contain stdout).

HowTo: Escape the Devillish XML



Avoid parsing XML yourself, make use of TectoMT API for reading.

1. Implement a simple block to print information to stdout.
2. Submit parallel printing, e.g.:

```
tools/cluster_utils/qrunblocks \  
  filelist \  
  "Print::Factored" \  
  --jobs 20 --no-save \  
  --join \  
  > joined_output
```

- `--no-save` avoids saving TMT files,
- `--join` waits for all the jobs to succeed and joins their stdouts preserving file order.

Caveats: NFS is the Bottleneck



`qrunblocks` simply splits the filelist and submits the jobs.

⇒ too many jobs accessing the same NFS server cause delays.

Current workarounds:

- Reduce the number of jobs.
- Spread your files to many NFS servers, e.g.:
`/net/cluster/COMPUTER/tmp/` for various computers
⇒ inefficient processing of non-local files.

Ultimate solution:

- Know which files are local to a node.
- Submit jobs only to nodes with unfinished files.
- Jobs themselves figure out which (local) files need to be processed.

Debugging Someone Else's Code



- Your particular data may crash some of the TectoMT blocks.
 - Debugging with huge datasets is slow or impossible.
 - Need to send a small bug report if unable to fix the bug yourself.
-

1. Find one of the problematic files (e.g. study qrunblocks logs).

2. Apply auto-diagnose:

```
$TMT_ROOT/tools/tests/auto_diagnose.pl --cleanup \  
file.tmt.gz targetdir 'block1 block2'
```

3. Run the test as instructed:

```
./targetdir/test.sh
```

Or simply send the targetdir to the assumed author.

Auto-diagnose finds the first crashing sentence, the first crashing block from the scenario, and construct a TMT file with just the sentence. The test.sh is just the command line to run the minimized test.

Suggested Applications



NLP hacking:

- Remove useless case markings, insert fake articles and preps:
Czech $\xrightarrow{\text{Perl}}$ Czenlish $\xrightarrow{\text{ISI ReWrite}}$ English (Cuřín, 2006)
- Move verbs to the end of the clause:
English $\xrightarrow{\text{TectoMT}}$ Hinglish $\xrightarrow{\text{Moses}}$ Hindi (Bojar et al., 2008b)
We needed ~230 lines of code, SVO→SOV alone is 12 lines.
- Truecasing based on names as marked by a lemmatizer/NER.

Feature fishing: Rich features for your favourite MT:

- Highlight non-local information, e.g. subject-verb agreement:
Cat...talked → *...talked+sg* vs. *Cats...talked* → *...talked+pl*

More details in Thursday and Friday lectures.

Summary



- TectoMT *can* be used on large data.
- Debugging is just a regular nightmare, not worse.

Suggested workflow for your TectoMT Project at Marathon:

1. Get a brilliant idea, find friends.
2. Adapt `tools/format_convertors` to load your input.
3. Setup your annotation scenario.
 - Add your own blocks for NLP hacking.
4. Use `qrunblocks` to annotate huge data.
5. Export to plaintext.
6. Train/apply/test your favourite MT system.

Ondřej Bojar, Miroslav Janíček, Zdeněk Žabokrtský, Pavel Češka, and Peter Beňa. 2008a. CzEng 0.7: Parallel Corpus with Community-Supplied Translations. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. ELRA.

Ondřej Bojar, Pavel Straňák, and Daniel Zeman. 2008b. English-Hindi Translation in 21 Days. In *Proceedings of the 6th International Conference On Natural Language Processing (ICON-2008) NLP Tools Contest*, Pune, India, December. NLP Association of India.

Jan Cuřín. 2006. *Statistical Methods in Czech-English Machine Translation*. Ph.D. thesis, ÚFAL, MFF UK, Prague, Czech Republic.

Petr Pajas and Jan Štěpánek. 2005. A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague Dependency Treebank 2.0. Technical Report TR-2005-29, ÚFAL MFF UK, Prague, Czech Rep.