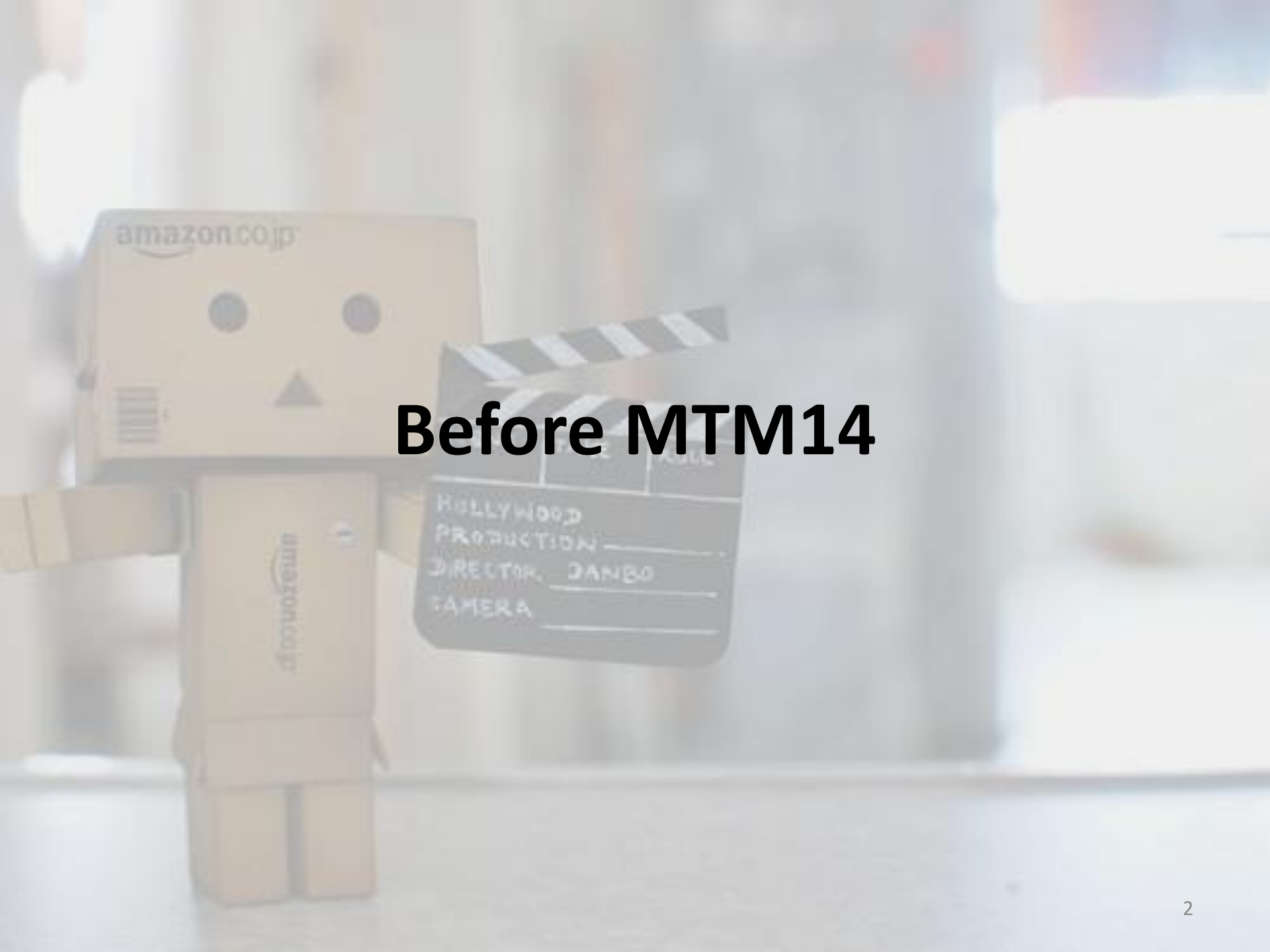
MT with NLTK
(yet another MT decoder)



Before MTM14

MTM14 Project Proposal

Everyone loves python

Everyone that loves python love NLTK

Everyone here loves MT

Let’s do MT with NLTK !!!

Improving NLTK support for statistical machine translation - Google Groups - Chromium

https://groups.google.com/forum/#!topic/nltk-dev/cPxFv_CA3vE

Google | Search for topics | +liling | Share

Groups | POST REPLY | 8 of 99+ (99+)

My groups
Home
Starred

Favourites

Click on a group's star icon to add it to your favourites

Recently viewed
- HackerspaceSG
- nltk-dev
- nltk
- Linux users IITM
- SemEval

Recent searches
- semeval 2016 (in seme…
- semeval3
- semeval
- uds-coli-2013 (in uds-c…
- cluster (in uds-coli-2011)

Recently posted to
- nltk-dev
- SemEval
- DSL Shared Task
- LxMLS-2014
- nltk-users

nltk-dev ›

Improving NLTK support for statistical machine translation

3 posts by 3 authors

Assign

Steven Bird Jul 14

Now that the first beta release of NLTK 3 has been finalized, it seems like a good time to think about new areas of functionality to include in NLTK. Of course, anyone can contribute new functionality at any time by submitting a pull request. However, there are some rather obvious gaps that would be good to fill.

Statistical machine translation occupies a substantial fraction of the ACL program, and there is a core body of knowledge represented by Philipp Koehn's textbook. By providing reference implementations for some SMT algorithms, corpus readers, and the like, NLTK could make this field more accessible.

We already have implementations for IBM models 1-3, the BLEU metric, and the Gale-Church aligner. It would be good to have decoders, and support for phrase-based models. As usual we are not looking for a high degree of optimisation, but rather a high degree of readability.

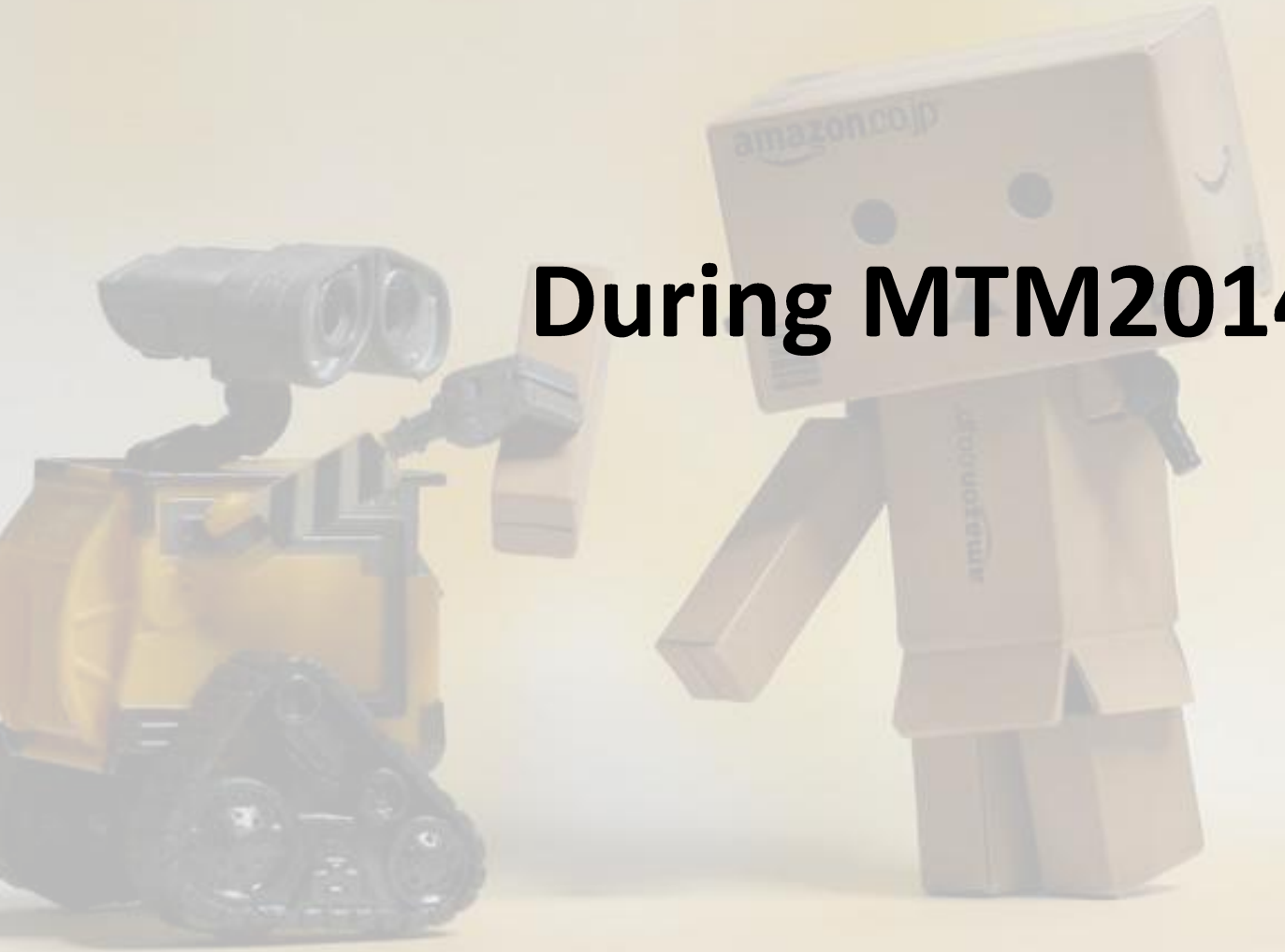
Would you like to volunteer to contribute something? Please just open an issue and add the "SMT" label.
https://github.com/nltk/nltk/issues/new

Does something already exist that we might be able to incorporate? Please add it to the following wiki page:
https://github.com/nltk/nltk/wiki/Machine-Translation

Other ideas... please just post them here and we can discuss.

Ewan Klein and I will be proposing further areas for new functionality in the coming weeks.

During MTM2014



Project Interim

- Discussing with NLTK devs
- Language Models too large to load
- Introduce new `translate` module in NLTK

Issues with implementing MT decoder in NLTK - Google Groups - Chromium

https://groups.google.com/forum/#!topic/nltk-dev/7fKRZV-HKYM

Google | Search for topics | +liling | Share

Groups | POST REPLY | 1 of 99+ (99+)

My groups
Home
Starred

Favourites

Click on a group's star icon to add it to your favourites

Recently viewed
- HackerspaceSG
- nltk-dev
- nltk
- Linux users IITM
- SemEval

Recent searches
- semeval 2016 (in seme…
- semeval3
- semeval
- uds-coli-2013 (in uds-c…
- cluster (in uds-coli-2011)

Recently posted to
- nltk-dev
- SemEval
- DSL Shared Task
- LxMLS-2014
- nltk-users

nltk-dev ›

Issues with implementing MT decoder in NLTK

8 posts by 4 authors

Assign

me (liling tan change) Sep 10

Dear NLTK devs,

Currently, we are at MTM14 and we're attempting some code sprints on MT functionality in NLTK and we're reaching the decoder part of the code where we need to discuss some issues with regards to the implementation.

Rather than building a simplistic implementation and then scaling it up with lots of hack, we decided that it's better to build something scalable to real data and still remain easy to understand/read. That way, it's easier to rewrite the code later for educational purposes.

There's several issues when it comes to decoders that native python might not be able to handle. And I hope we can discuss these issues here:

- Administratively, it would be kind of weird to put the decoder in the `nltk/align` library, should we create an MT directory in NLTK?
- Filtering the phrase table prior to decoding is some sort of issue, any suggestion to do this more efficiently?
- Implementing a native python readers for language models and phrase tables is bulky and it's not realistic to load it efficiently, people at MTM suggested KenLM wrapper which includes .cpp , .pyx and .pyd codes to be added to NLTK (from https://kheafield.com/code/kenlm/developers/) . That would mean that NLTK might have to ship with these files to run the decoder.
- Tuning the decoder would require some machine learning module and the main suggestion is to reimplement MERT with scikit-learn, would that be okay? or should we use make a more native python implementation?

Regards,
Liling

Click here to Reply

me Alternatively, we are also considering porting code from Kriya, would that be okay to port/fork code from another python project into NLTK? Sep 10

dimazest Hi, I'm not a cure developer, but as a user I think that NLTK should be able to scale to "real data". Even if the code is not extremely easy to understand. To put it in other Sep 10

Project Report

PyDev - nltk/nltk/translate/decoder-old.py - Eclipse Platform

```
86 '''
87 filedir = '/media/alvas/046f32c0-39fa-44e6-85b1-d8c2e3a661dc/'
88 langmodelfile = filedir + 'lm.zh.arpa.gz' # 28,465,057 ngrams, 290 secs
89 phrasetablefile = filedir + 'phrase-table.gz' # 21,873,720 phrase pairs, 120 secs
90 sent = "ロボット 業界 における 環境 問題 への取り組み 環境 問題 への取り組み".split()
91 '''
92
93 # wget http://www.statmt.org/moses/download/sample-models.tgz
94 testfile = 'test.in'
95 phrasetablefile = 'phrase-table.gz'
96 langmodelfile = 'europarl.srilm.gz'
97
98 sent ='das ist ein kleines haus'.split()
99
100 print "Loading phrase table...\n(you can grab a cup of coffee)\n"
101 tm = read_phrase_table(phrasetablefile)
102 print "Loading language model... \n(you can go take a walk before coming back)\n"
103 lm = LanguageModel(langmodelfile)
104 print "Finish loading models\n"
105
106 print "Decoding:", sent
107 wiener = monotone_stack_decode(sent, tm, lm)
108 print wiener
109 print schnitzel(wiener)
110
```

```
<terminated> /home/alvas/git/nltk/nltk/translate/decoder-old.py
Loading phrase table...
(you can grab a cup of coffee)

Loading language model...
(you can go take a walk before coming back)

Finish loading models

Decoding: ['das', 'ist', 'ein', 'kleines', 'haus']
hypothesis(logprob=-8.4053796, lm_state=('small', 'house'), predecessor=hypothesis(logprob=-2.6311216, lm_state=('a', 'small'), predecessor=hypothesis(logprob=-0.37
this is a small house
```

Project Report

- Added monotone stack decoder to `translate` in NLTK
- Successfully decoded:

'*das ist ein kleines haus*' -> '*this is a small house*'

Post MTM

- Cleaned and documented the code
- Pull – Merge to main dev branch (https://github.com/nltk/nltk/pull/772)

Fin.