The WeSearch Corpus, Treebank, and Treecache:
A Comprehensive Sample of User-Generated Content

Jonathon Read♣, Dan Flickinger♡, Rebecca Dridan♣,
Stephan Oepen♣, and Lilja Øvrelid♣
♣ University of Oslo, Department of Informatics
♡ Stanford University, Center for the Study of Language and Information

{jread|rdridan|oe|liljao}@ifi.uio.no, danf@stanford.edu

Introduction

User-generated content

- Potentially a rich source of information:
  - IE/QA using the 'wisdom of the masses'
  - Opinion mining
  - NLP for the social sciences
- But language technology can struggle with informal content (Foster *et al*. 2011)

Introduction

User-generated content

- Potentially a rich source of information:
  - IE/QA using the ‘wisdom of the masses’
  - Opinion mining
  - NLP for the social sciences
- But language technology can struggle with informal content (Foster *et al*. 2011)
- WeSearch Data Collection

Introduction

Organised with respect to:

Domain: similarities of **content**

Genre: similarities of **form**

Introduction

Organised with respect to:

Domain: similarities of **content**

Genre: similarities of **form**

Three components:

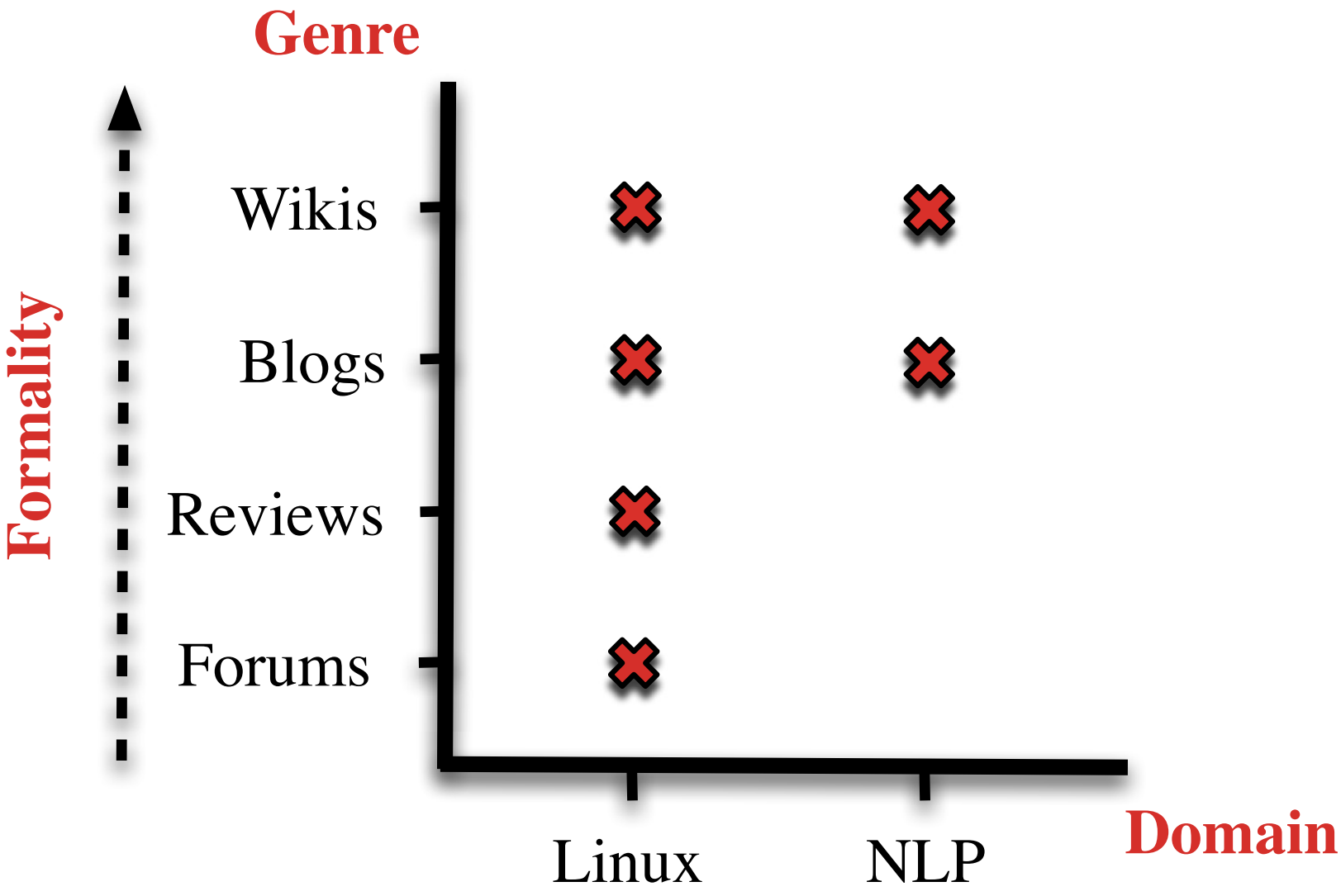
Corpus: unannotated text

Treebank: gold-standard annotations

Treecache: automatically-generated annotations

Outline

Data Selection



Data Selection

Domain	Genre	Source(s)	Format
NLP	Wikipedia	WeScience (Ysterøl *et al.* 2009)	Wikitext

NLP	Blogs	`blog.cyberling.org`
		`gameswithwords.fieldofscience.com`
		`lingpipe-blog.com`
		`nlpers.blogspot.com`
		`thelousylinguist.blogspot.com`
			HTML

Linux	Wikipedia	`www.wikipedia.org`	Wikitext

Linux	Blogs	`embraceubuntu.com`
		`www.linuxscrew.com`
		`www.markshuttleworth.com`
		`www.ubuntugeek.com`
		`ubuntu.philipcasey.com`
		`www.ubuntu-unleashed.com`
			HTML

Linux	Software reviews	`www.softpedia.com/reviews/linux/`	HTML
Linux	User forums	The 'Unix & Linux' subset of the April 2011 Stack Exchange Creative Commons Dump.	HTML

Harvesting and Extraction

Blogs and reviews are in 'wild' HTML

1. Regular expression for **title** in HTML header
2. Regular expression for the **start tag** of the body
3. Function to **remove superfluous text**

Forum data is sanitised HTML from Stack Exchange.

Harvesting and Extraction

Further cleaning for blogs, reviews and forums:

- Removal of tables and comments
- Placeholders for `img` and `code`
- Sentence segmentation using the Stanford CoreNLP Tools
- Force segmentation following sentence-breaking tags

Haversting and Extraction

For Wikipedia we follow Ysterøl *et al*. (2009):

- Seed set gathered from Linux sets in the category system
- The collection is grown using a link analysis
- Sentence segmentation with tokeniser

Corpus Organisation

	Content	Markup	Organisation
L0	Raw source files	all	source site
L1	Cleaned utterances	selected	sections
L2	Cleaned utterances	normalised	sections

L1 and L2 also includes:

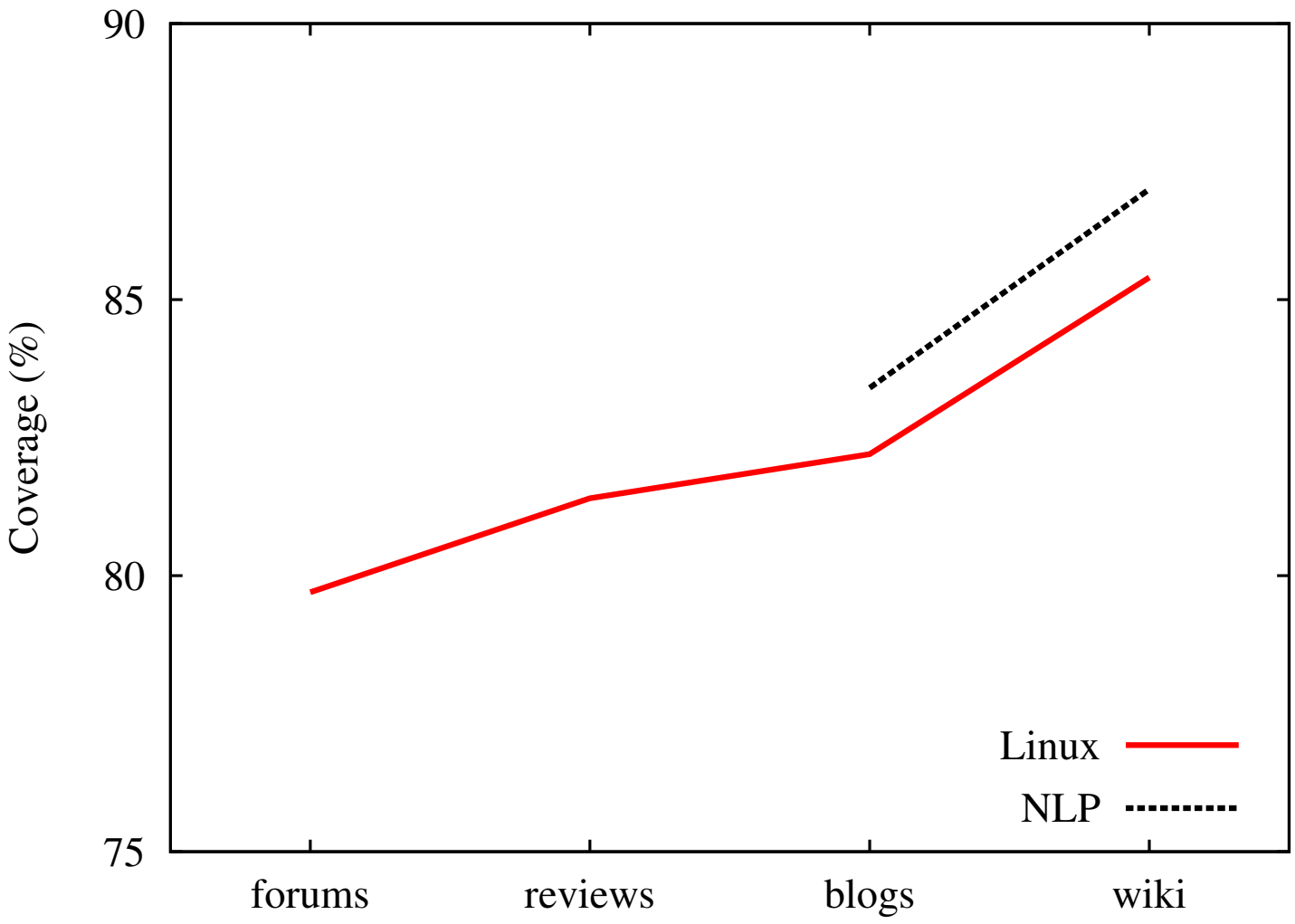
- pointers from items to the raw source, with character offset
- character offsets that represent deletions from the source

Initial Parsing Results

Parsing using the **English Resource Grammar** with **PET**.

Domain	Genre	Items	Length	Coverage	Exhaustion
NLP	Wiki	10,577	14.5	87.0%	7.4%
NLP	Blog	36,104	17.1	83.4%	5.8%
Linux	Wiki	37,263	18.5	85.4%	9.6%
Linux	Blog	57,599	12.9	82.2%	4.1%
Linux	Review	9,667	19.1	81.4%	5.6%
Linux	Forum	53,160	14.5	79.7%	2.9%

Initial Parsing Results



Initial Parsing Results

(*)? “Apparently this prevented it from being found by which.”

Initial Parsing Results

(*)? "Apparently this prevented it from being found by which."

"Apparently this prevented it from being found by `which`."

WHICH(1)

```
NAME
     which - shows the full path of (shell) commands.

SYNOPSIS
     which [options] [--] programname [...]

DESCRIPTION
     Which takes one or more arguments.  For each of its arguments it prints
     to stdout the full path of the executables that would have been exe-
     cuted when this argument had been entered at the shell prompt.  It does
     this by searching for an executable or script in the directories listed
     in the environment variable PATH using the same algorithm as bash(1).
```

Initial Treebanking Reflections

Taking one section from the NLP-Blogs collection:

- Manually-corrected sentence segmentation
  - 994 automatic items to 1,078 manual items
  - End of sentence obscured by markup
    e.g. “Find out more here.”
  - . . . but introduces breaks on file names *etc.*
    e.g. `init.d`. `readme.txt`
  - Unusual use of punctuation, e.g. Yahoo!
  - Mishandled abbreviations, e.g. Ph. D.

Initial Treebanking Reflections

Taking one section from the NLP-Blogs collection:

- Manually-corrected sentence segmentation
  - 994 automatic items to 1,078 manual items
  - End of sentence obscured by markup
    e.g. "Find out more here."
  - . . . but introduces breaks on file names *etc.*
    e.g. `init.d`. `readme.txt`
  - Unusual use of punctuation, e.g. Yahoo!
  - Mishandled abbreviations, e.g. Ph. D.

- Treebanking
  - Omissions from the ERG lexicon:
    - emoticons, e.g. :), :P
    - exclamations, e.g. D'oh, ah-ha
    - abbreviations, e.g. btw, omg, imho
  - Genre-specific informal expressions, e.g.
    "the likes of [...]", and "crammed in some [...]"

Outlook

- Using the WDC:
  - Evaluating parser adaptation
  - Intepreting markup
- Refining the WDC:
  - Segmentation
  - Additional genres and domains

Outlook

- Using the WDC:
  - Evaluating parser adaptation
  - Intepreting markup
- Refining the WDC:
  - Segmentation
  - Additional genres and domains

WeSearch Data Collection

`www.delph-in.net/wesearch`