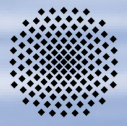


University of Stuttgart
Germany

Institute for Natural Language Processing

Bootstrapping Semantic Lexicons for Technical Domains

Patrick Ziering[1] Lonneke v. d. Plas[1] Hinrich Schütze[2]

[1]Institute for Natural Language Processing (IMS), University of Stuttgart

[2]Center for Information and Language Processing (CIS), University of Munich

IJCNLP 2013

Overview

Lexicon Bootstrapping and issues in technical domains

Basilisk for the patent domain
- Basilisk-G(eneral)
- Basilisk-C(oordination)

Experiment
- Setup
- Evaluation methodology
- Results

Conclusion

LEXICON BOOTSTRAPPING AND ISSUES IN TECHNICAL DOMAINS

Lexicon Bootstrapping

- One of the most-widely used ways for lexicon acquisition
- A seed list is iteratively expanded by adding the most reliable (highest-ranked) words in each iteration
- The ranking of words is usually based on context patterns that surround a word (e.g., *made of <X>*)
- Scoring of words and context patterns is based on frequency

Lexicon Bootstrapping

- One of the most-widely used ways for lexicon acquisition
- A seed list is iteratively expanded by adding the most reliable (highest-ranked) words in each iteration
- The ranking of words is usually based on context patterns that surround a word (e.g., *made of <X>*)
- Scoring of words and context patterns is based on frequency
- We will show:
  definition and extraction of words and their context patterns is a critical factor in lexicon bootstrapping for technical domains

Lexicon Bootstrapping

- We adopted the basic architecture and scoring functions of Basilisk (Thelen and Riloff, 2002)

```
1: lexicon ← seed
2: for int i = 0; i < m; i++ do
3:     patterns ←patternsOf(lexicon)
4:     score(patterns)
5:     patterns ← return-top-k(patterns,20 + i)
6:     terms ← termsOf(patterns) − lexicon
7:     score(terms)
8:     lexicon ← lexicon ∪ return-top-k(terms,t)
9: end for
10: return lexicon
```

Issues in technical domains

Problems of original Basilisk for technical domains:

- Basilisk uses lexico-syntactic patterns
  (e.g., <*subj*> *passive verb* → <subj> was acquired)

  These patterns
  - are based on a shallow parser
  - are numerous
  - frequently lack selectivity

- Basilisk induces lexicons of only single words

Issues in technical domains

Lexico-syntactic patterns are based on a shallow parser

- Parsing patents takes much longer than parsing standard data
- Syntax in patent text is complex and characterized by long clauses

Issues in technical domains

Lexico-syntactic patterns are numerous

- Working with millions of patterns (from our large patent corpus) leads to slow Basilisk performance in later iterations
- The size of induced lexicons is limited

Issues in technical domains

Lexico-syntactic patterns frequently lack selectivity

- For poorly represented semantic classes (e.g., DISEASE):

  a lexico-syntactic pattern with less discriminatory power is ranked high in later iterations
  $\rightarrow$ semantic drift

- The pattern *treatment of <X>* can hold a slot for both DISEASE and other entity classes like PERSON
  - *treatment of cancer*
  - *treatment of prisoners*

Issues in technical domains

- Original Basilisk induces lexicons of only single words
  - While restricting to head nouns works acceptably for general corpora, it misses too many terms in technical domains
  - The class SUBSTANCE comprises about 45% MWEs in patents (e.g., *alkyl trimethyl ammonium methosulfate*)
- For these reasons, we decided to adapt Basilisk to the requirements of large technical corpora such as patents

BASILISK FOR THE PATENT DOMAIN

Basilisk for the patent domain

We used two kinds of Basilisk patterns:

- General patterns
  - A scalable generalization of lexico-syntactic patterns
- Coordination patterns
  - A well-known context pattern in lexical acquisition
  - Predestined for lexicon bootstrapping in large patent corpora

For these pattern types, we extract both words and MWEs

Basilisk-G(eneral)

- A Basilisk version with general patterns for entity types that are poorly represented in coordinations
- A Basilisk baseline for large patent corpora

- A simple chunker that identifies NPs using a PoS pattern
- General patterns are defined using the PoS pattern:
  $w_{-i} \ldots w_{-1}$ <NP> $w_1 \ldots w_j$
  where $0 \leq i, j \leq 3$ and $i + j \geq 2$
  and $\exists k[PoS(w_k) \in \{noun, verb\}]$

Basilisk-G(eneral)

- Covers most of original Basilisk's lexico-syntactic patterns
- Also covers lexical contexts like fragments of coordinations:
  , SILVER, <NP> OR PLATINUM
- Avoidance of syntactic parsing of patent text
- Frequency thresholds for terms, patterns and term-pattern combinations

Basilisk-C(oordination)

Some advantages of using coordination patterns:

- Strong semantic coherence: coordinated terms are often co-hyponyms of a common semantic class
- Coordination size in patents is frequently greater than 2
  - General or lexico-syntactic patterns have arity 1 (this is frequently not restrictive enough)

  - *treatment of <X>*
  - *congenital heart defect, atherosclerosis, <X>, scleroderma or tuberous sclerosis*

Basilisk-C(oordination)

Some advantages of using coordination patterns:

- No need for syntactic parsing
  - Coordinations can be extracted reliably with PoS patterns
- Restriction to subset of patterns
  - Limits the overproduction of patterns radically
  - Application to the largest domain-specific corpus ever used for semantic bootstrapping
  - Induction of very large lexicons
- Easy way of extracting MWEs
  - Connectors (commas, conjunctions, etc.) are used as MWE boundaries

EXPERIMENT

Setup

Data:

- Patent descriptions of 561K English patents filed at the European Patent Office (EPO)
- Sentence splitting and tokenization:
  $\rightarrow$ about 4.6 billion tokens
- Lemmatization and PoS tagging
- A subset of 25,000 sentences up to 100 tokens is parsed

Setup

Data:

- Basilisk-G
  - 1.6 billion unique general patterns

    ↓ *Frequency thresholds*

    56 million unique general patterns

- Basilisk-C
  - 9.7 million unique coordinations

Setup

Semantic classes:

- SUBSTANCE
  - a particular kind of physical matter with uniform properties (e.g., *gold*, *wood*, *air*, *polyvinyl chloride*, . . .)
  - prevalent in patent domain; no semantic drift
  - Seed: 4223 substances (WordNet SuperSense)

- DISEASE
  - an abnormal condition that affects the body of an organism (e.g., *AIDS*, *multiple sclerosis*, *prostatitis*, . . .)
  - infrequent in patent domain; semantic drift
  - Seed: 239 diseases (Simple English Wikipedia)

Evaluation methodology

Evaluation of bootstrapped lexicons on semantic tagging task

- This evaluation considers:
  - term frequency (frequent terms have a higher impact)
  - ambiguity (ambiguous terms with a rare class sense depress tagging accuracy)
- Tagging method – lexicon lookup:
  - For a specialized class and domain, ambiguity of terms is a limited phenomenon (Quadir and Riloff, 2012)
- Gold standard:
  - 2000 patent sentences for SUBSTANCE and DISEASE annotations
  - macro-κ: .712
    micro-κ: .818

Results

size	SUBSTANCE					
	P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C
seed	.597		.491		.539	
5000	.599*	.598	.494*	.492*	.542*	.540*
10,000	.605*	.604*	.502*	.504*	.549*	.549*
20,000	.610*	.614*	.509*	.529*†	.555*	.568*†
40,000	.612*	.619*	.515*	.549*†	.559*	.582*†

- We evaluate the seed lexicon (baseline) and lexicons of size $5K, 10K, 20K$ and $40K$

Results

size	SUBSTANCE					
	P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C
seed	.597		.491		.539	
5000	.599*	.598	.494*	.492*	.542*	.540*
10,000	.605*	.604*	.502*	.504*	.549*	.549*
20,000	.610*	.614*	.509*	.529*†	.555*	.568*†
40,000	.612*	.619*	.515*	.549*†	.559*	.582*†

- All measures outperform the baseline
- Starred performance numbers are significantly higher than the number above it

Results

size	SUBSTANCE					
	P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C
seed	.597		.491		.539	
5000	.599*	.598	.494*	.492*	.542*	.540*
10,000	.605*	.604*	.502*	.504*	.549*	.549*
20,000	.610*	.614*	.509*	.529*†	.555*	.568*†
40,000	.612*	.619*	.515*	.549*†	.559*	.582*†

- Basilisk-C outperforms Basilisk-G in most cases
- Performance numbers with a dagger (†) show a significant superiority to Basilisk-G

Results

size	SUBSTANCE					
	P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C
seed	.597		.491		.539	
5000	.599*	.598	.494*	.492*	.542*	.540*
10,000	.605*	.604*	.502*	.504*	.549*	.549*
20,000	.610*	.614*	.509*	.529*†	.555*	.568*†
40,000	.612*	.619*	.515*	.549*†	.559*	.582*†

- SUBSTANCE is a prevalent class in patents
  - No semantic drift (precision decline) at these lexicon sizes
- Basilisk-C clearly outperforms Basilisk-G in recall
  - Closer inspection: Spurious tokens in Basilisk-G terms
  - Coordinations yield a better MWE segmentation

Results

size	DISEASE					
	P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C
seed	.793		. 233		.360	
5000	.790	.724	.455*	.556*†	.578*	.629*†
10,000	.476	.643†	.645*	.602*	.548	.622†
20,000	.392	.530†	.642	.701*†	.487	.604†
40,000	.300	.473†	.642	.720†	.409	.571†

- All recall and F_1 measures outperform the baseline

Results

size	DISEASE					
	P		R		F_1	
	B-G	B-C	B-G	B-C	B-G	B-C
seed	.793		. 233		.360	
5000	.790	.724	.455*	.556*†	.578*	.629*†
10,000	.476	.643†	.645*	.602*	.548	.622†
20,000	.392	.530†	.642	.701*†	.487	.604†
40,000	.300	.473†	.642	.720†	.409	.571†

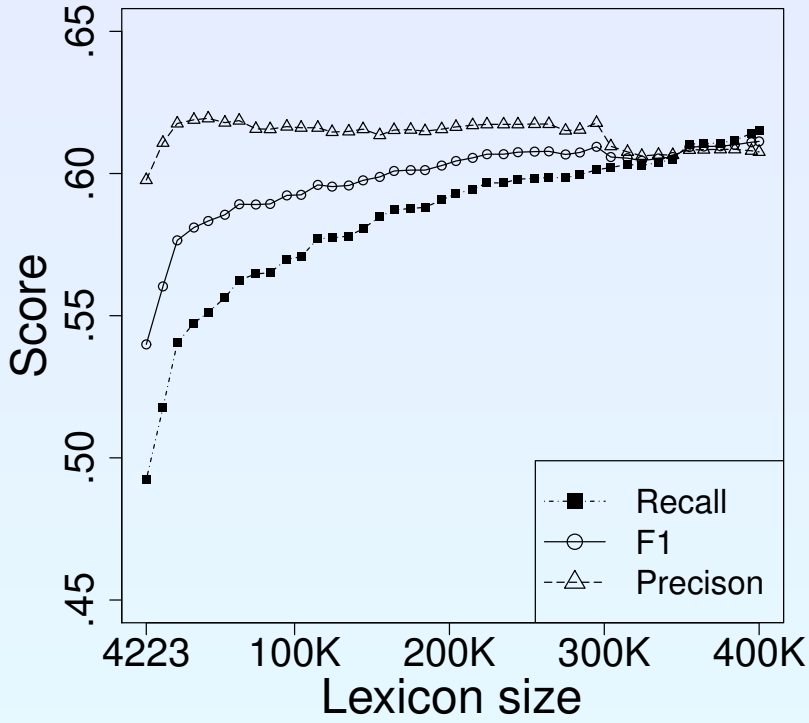
- Precision decreases rapidly
  - Closer inspection: semantic drift to technical properties
- Altogether Basilisk-C outperforms Basilisk-G in precision
  - Coordinations remedy semantic drift in smaller classes

Results

- We cannot run original Basilisk (and Basilisk-G) for very large lexicons ($\geq 40K$)
- Scalability benefit of Basilisk-C: fewer very strong patterns
- We are able to create large SUBSTANCE lexicons up to size $400K$ using Basilisk-C

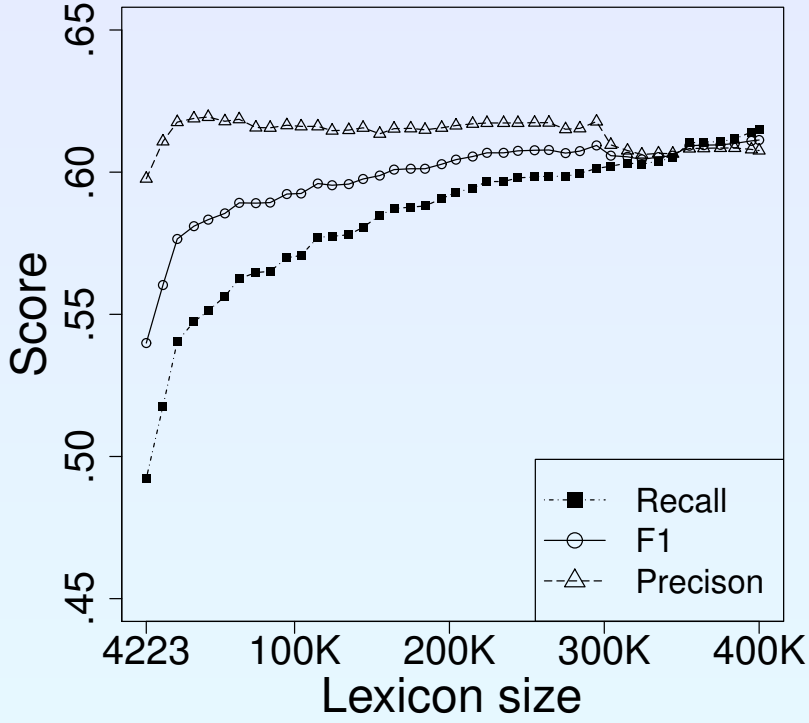
Very large lexicons can be induced using coordinations

Results



- Basilisk-C performance as a function of lexicon size
- Upward trend for recall and F_1 up to very large lexicons

Results



Best Basilisk-C performance at about 400K:

Precision	Recall	F_1
.608	.615	.611

Results on EPO subcorpus

- Comparison with original Basilisk:
  - Lexicon quality of 20,000 substances
- Parsing of 25,000 EPO sentences up to 100 tokens
- Original Basilisk only regards head nouns
- We add a modified Basilisk with MWE recognition
  - Basilisk-LS$_{head}$: lexico-syntactic patterns for head nouns
  - Basilisk-LS$_{MWE}$: lexico-syntactic patterns for basic NPs

Results on EPO subcorpus

System	P	R	F_1
Basilisk-LS$_{head}$	.437	.502	.467
Basilisk-LS$_{MWE}$	.582	.506	.541
Basilisk-G	.560	.524	.542
Basilisk-C	.620	.567	.592

- Poor precision when tagging only head nouns

Results on EPO subcorpus

System	P	R	F_1
Basilisk-LS$_{head}$	.437	.502	.467
Basilisk-LS$_{MWE}$	.582	.506	.541
Basilisk-G	.560	.524	.542
Basilisk-C	.620	.567	.592

- Basilisk-LS$_{MWE}$ and Basilisk-G only slightly outperform the seed baseline ($F_1 = .539$)
- Basilisk-C works outstandingly when tagging substances in patents

CONCLUSION

Conclusion

- Basilisk pattern adaptation for large technical corpora:
  - Basilisk-G(eneral)
  - Basilisk-C(oordination)
- Both systems outperform:
  - seed baseline
  - original Basilisk (Basilisk-LS)
- Basilisk-LS$_{MWE}$ and Basilisk-G perform similarly (F_1)
- Basilisk-C clearly outperforms Basilisk-G due to coordination benefits:
  - \+ Coordinations yield a better MWE segmentation
  - \+ Coordinations remedy semantic drift in smaller classes
  - \+ Very large lexicons can be induced using coordinations