

Using n-gram statistics for grammatical correction and entity recognition

Gianluca Demartini

University of Sheffield, UK

gianlucademartini.net

Research Interests

- **Entity-centric Information Access (2005-now)**
 - Structured/Unstruct data (SIGIR 12), TRank (ISWC 13)
 - NER in Scientific Docs (WWW 14), Prepositions (CIKM 14)
- **Hybrid Human-Machine Systems (2012-now)**
 - ZenCrowd (WWW 12, VLDBJ), CrowdQ (CIDR 13)
 - Human Memory based Systems (WWW 14, PVLDB)
- **Better Crowdsourcing Platforms (2013-now)**
 - Pick-a-Crowd (WWW 13), Malicious Workers (CHI 15)
 - Scale-up Crowdsourcing (HCOMP 14), Dynamics (WWW 15)
 - EPSRC First Grant 2016-2018

Grammatical Correction

Motivations and Task Overview

- Grammatical correction is important by itself
 - Also as a part of Machine Translation or Speech Recognition

Correction of textual content written by English Learners.

I am new ~~in~~ android programming.
[to, at, for, ...]

⇒ Rank candidate prepositions by their likelihood of being correct in order to potentially replace the original.

What we do

- English language only
 - Standard collection: CoNLL-2013
 - New collection based on Web user-generated content: Stack Exchange
- Preposition correction (13% of all errors) at sentence level
- N-gram decomposition of the input sentence
- Ranking of prep by the likelihood of being correct
- Define features and binary classify each prep

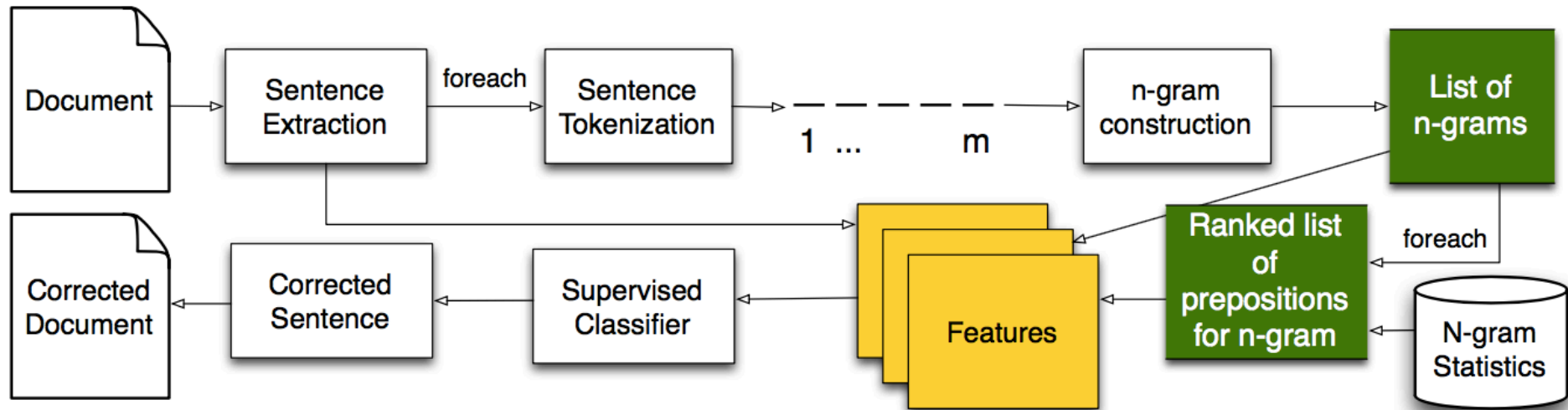
Key Ideas

- Usage of a particular preposition is governed by a particular word/n-gram;

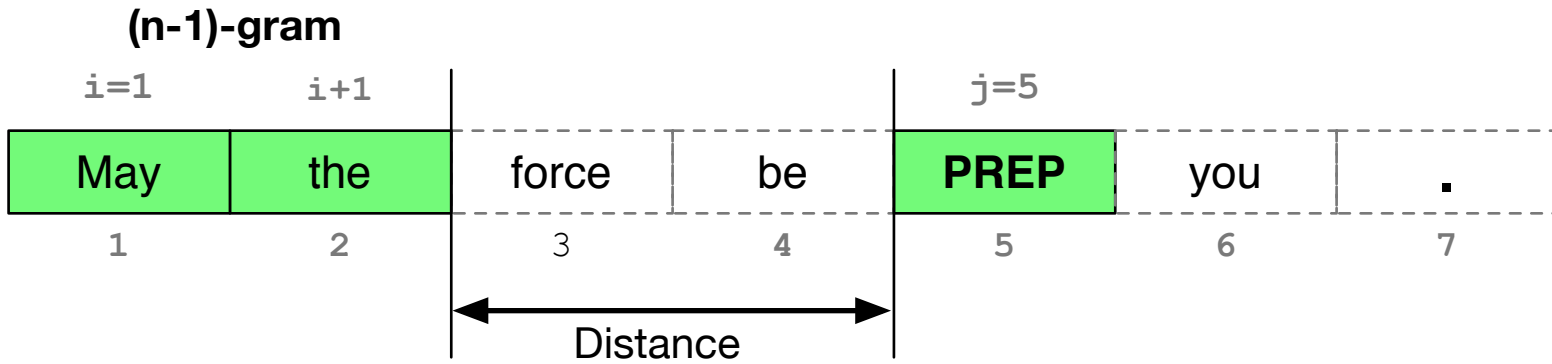
⇒ Task: select/aggregate n-grams that influence preposition usage;

⇒ Use n-gram association measures to score each preposition.

Processing Pipeline



Tokenization and n-gram distance



$$\min(|i-j|, |i+n-2-j|)$$

N-gram	Type	Distance
the force PREP	3gram	-2
force be PREP	3gram	-1
be PREP you	3gram	0
PREP you .	3gram	1

N-gram	Type	Distance
be PREP	2gram	-1
PREP you	2gram	1
PREP .	2gram	2

N-gram association measures

Motivation:

use association measures to compute a score that will be proportional to the likelihood of an n-gram appearing together with a preposition.

N-gram	PMI scores by preposition
force be PREP	(with: -4.9), (under: -7.86), (at: -9.26), (in: -9.93), ...
be PREP you	(with: -1.86), (amongst: -1.99), (beside: -2.26), ...
PREP you .	(behind: -0.71), (beside: -0.82), (around: -0.84), ...

Background N-gram collection: **Google Books N-grams.**

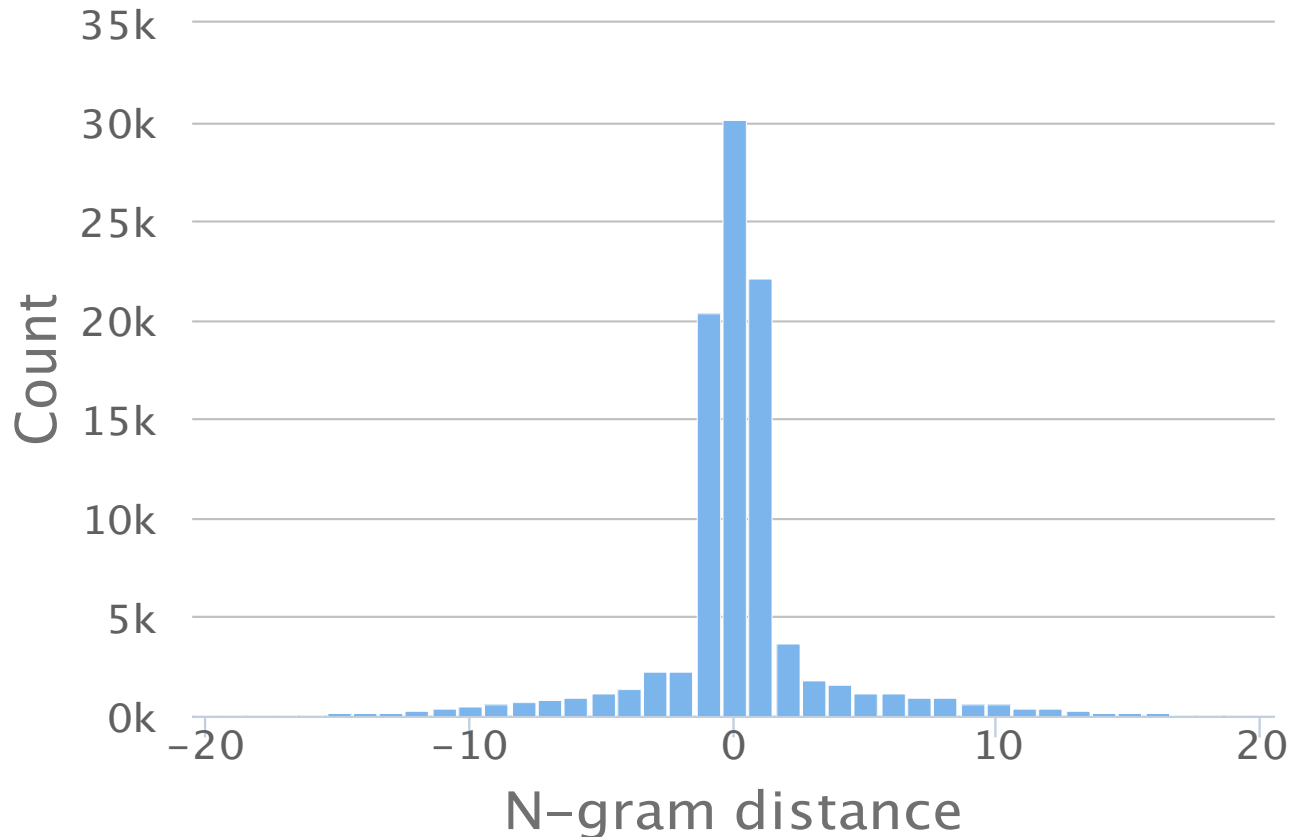
PMI-based Features

- **Average rank** of a preposition among the ranks of the considered n-grams;
- **Average PMI score** of a preposition among the PMI scores of the considered n-grams;
- **Total number of occurrences** of a certain preposition on the **first position in the ranking** among the ranks of the considered n-grams.

Calculated across 2 logical groups (considered n-grams):

- N-gram size;
- N-gram distances.

Central N-grams



Distribution of correct preposition counts on top of PMI rankings with respect to n-gram distance.

Other features

- Confusion matrix values

	to	in	of	for	on	at	with	from
to	0.958	0.007	0.002	0.011	0.004	0.003	0.005	0.002
in	0.037	0.79	0.01	0.009	0.066	0.036	0.015	0.008

- Some prep are most likely correct ('but' 0.992)
- POS tags: 5 most frequent tags + "OTHER" catch-all tag;
- Preposition itself: sparse vector of the size of the candidate preposition set.

Preposition selection

Supervised Learning algorithm.

- Two-class classification with a confidence score for every preposition from the candidate set;
- Every candidate preposition will receive its own set of feature values;

Classifier: random forest.

Errors are 5%. Balancing by under-sampling non-errors.

Training/Test Collections

Training collection:

- First Certificate of English (Cambridge exams)

Test collections:

- CoNLL-2013 (50 essays written by NUS students)
- StackExchange (historical edits)

	Cambridge FCE	CoNLL-2013	StackExchange
N# sentences	27k	1.4k	6k

Experiments: Feature Importance

Feature name	Importance score
Confusion matrix probability	0.28
Top preposition counts (3grams)	0.13
Average rank (distance=0)	0.06
Central n-gram rank	0.06
Average rank (distance=1)	0.05

All top features except “confusion matrix” are based on the **PMI scores**.

Test Collection Evaluation

Collection	Approach	Precision	Recall	F1 score
CoNLL-2013	NARA Team @CoNLL2013	0.2910	0.1254	0.1753
	N-gram-based classification	0.2592	0.3611	0.3017
StackExchange	N-gram-based classification	0.1585	0.2185	0.1837
	N-gram-based classification (cross-validation)	0.2704	0.2961	0.2824

Takeaways

- PMI association measures
 - + preposition ranking
- ⇒ allow to significantly outperform the state of the art.
- Portable approach (train on one collection to test on a different one)

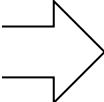
Named Entity Recognition

Problem Definition

1. INTRODUCTION

Nowadays, accessing information on the Internet through search engines has become a fundamental life activity. Current web search engines usually provide a ranked list of URLs to answer a query. This type of information access does a good job for dealing with simple navigational queries by leading users to specific websites. However, it is becoming increasingly insufficient for queries with vague or complex information need. Many queries serve just as the start of an exploration of related information space. Users may want to know about a topic from multiple aspects. Organizing the web content relevant to a query according to user intents would benefit user exploration. In addition, a list of URLs couldn't directly satisfy user information need. Users have

Entity type:
scientific concept

- 
- search engine
 - web search engine
 - navigational query
 - user intent
 - information need
 - web content
 - ...

Roman Prokofyev, Gianluca Demartini, and Philippe Cudré-Mauroux. **Effective Named Entity Recognition for Idiosyncratic Web Collections**. In: 23rd International Conference on World Wide Web (WWW 2014).

Traditional NER

Types:

- Maximum Entropy (Mallet, NLTK)
- Conditional Random Fields (Stanford NER, Mallet)

Properties:

- Require extensive training
- Usually domain-specific, different collections require training on their domain
- Very good at detecting such types as Location, Person, Organization

Proposed Approach

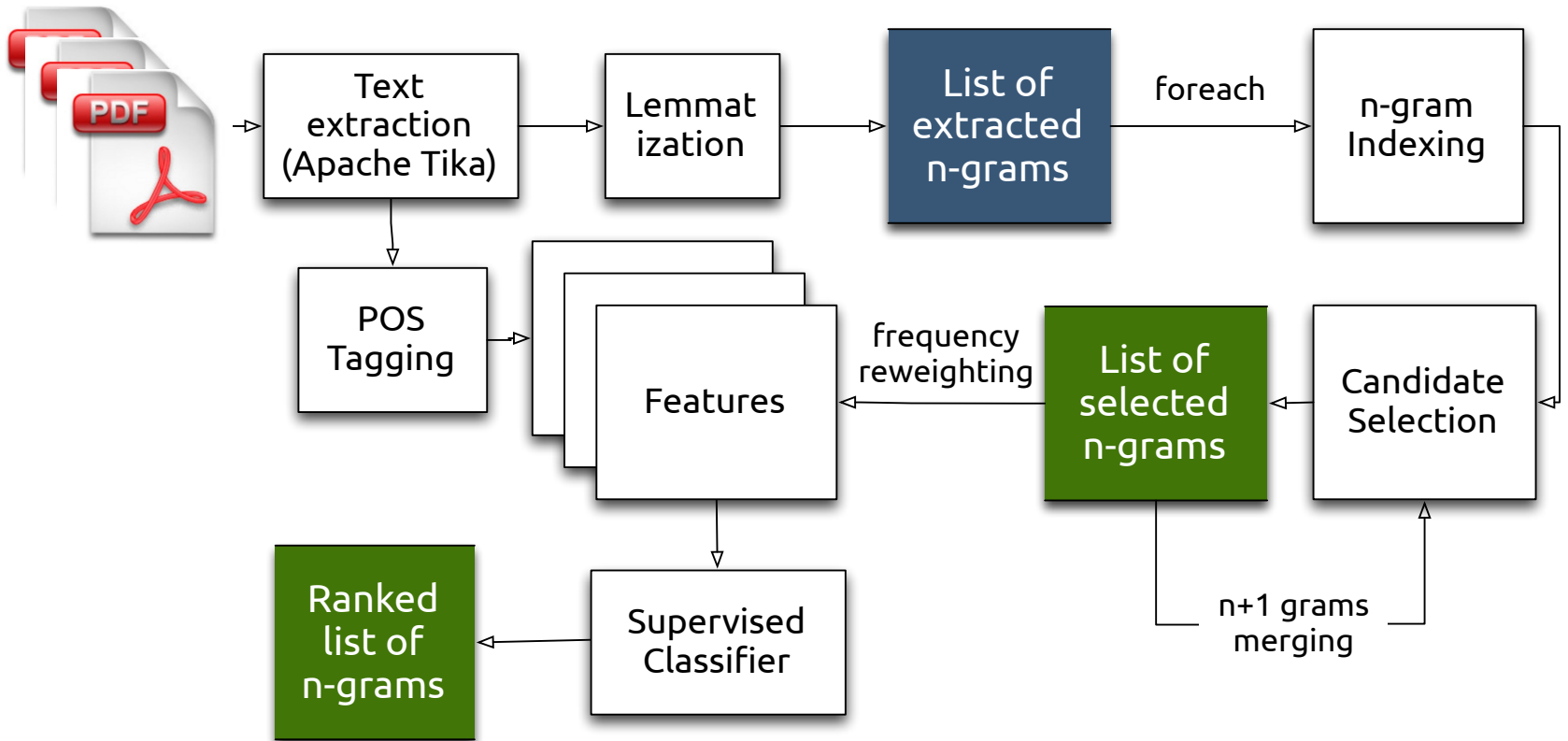
Our problem is defined as a classification task.

Two-step classification:

- Extract candidate named entities using frequency filtration algorithm.
- Classify candidate named entities using supervised classifier.

Candidate selection should allow us to greatly reduce the number of n-grams to classify, possibly without significant loss in Recall.

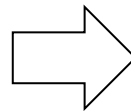
Pipeline



Candidate Selection: Part I

Consider all bigrams with frequency $> k$
($k=2$):

candidate named:	5
entity are:	4
entity candidate:	3
entity in:	18
entity recognition:	12
named entity:	101
of named:	10
that named:	3
the named:	4



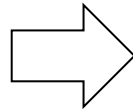
candidate named:	5
entity candidate:	3
entity recognition:	12
named entity:	101

NLTK stop word filter

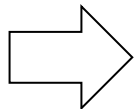
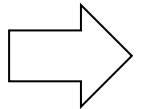
Candidate Selection: Part II

Trigram frequency is looked up from the n-gram index.

```
candidate named:      5
entity candidate:     3
entity recognition:  12
named entity:         101
```



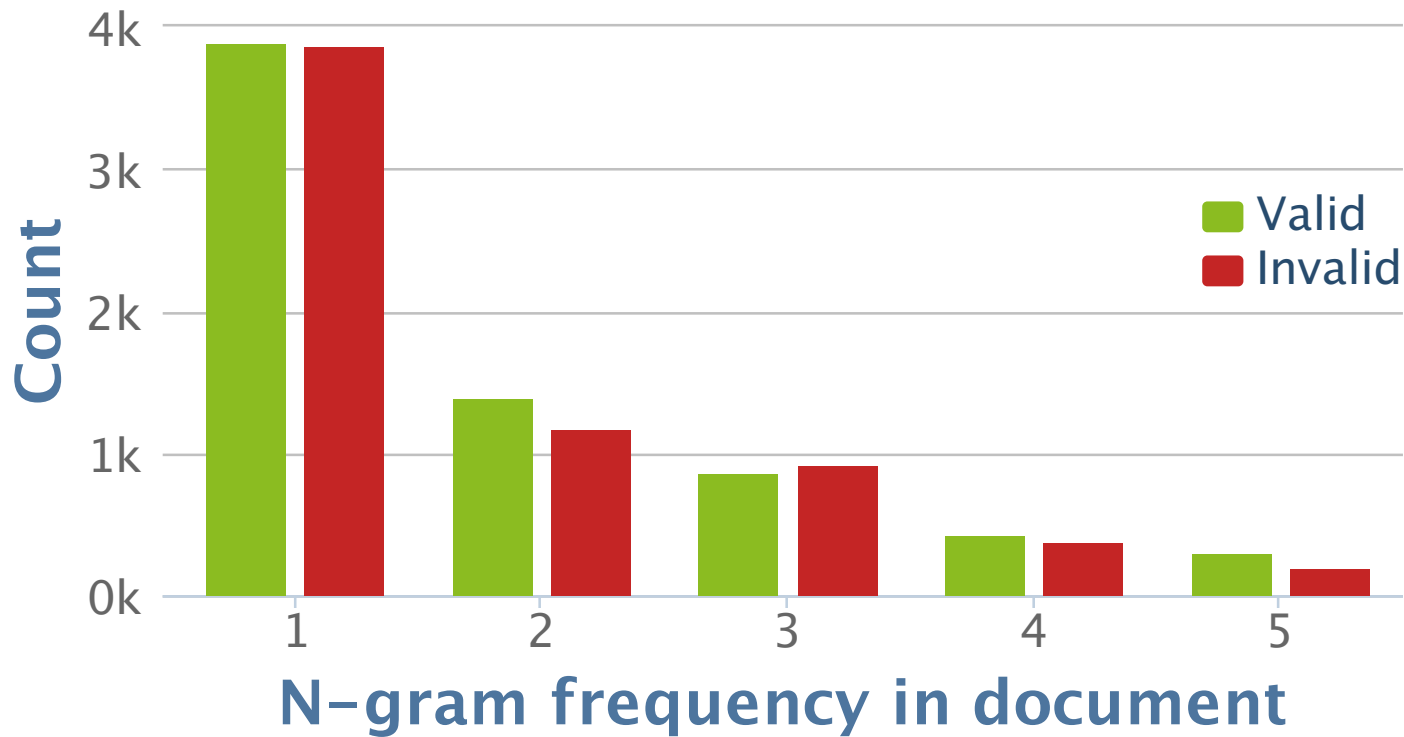
```
candidate named entity: 5
named entity candidate: 3
named entity recognition: 12
named entity: 101
candidate named: 5
entity candidate: 3
entity recognition: 12
```



```
candidate named entity: 5
named entity candidate: 3
named entity recognition: 12
named entity: 81
candidate named: 0
entity candidate: 0
entity recognition: 0
```


Candidate Selection: Discussion

Possible to extract n-grams ($n > 2$) with frequency $\leq k$



After Candidate Selection

ABSTRACT

Many private and/or public organizations have been reported to create and monitor **targeted Twitter streams** to collect and **understand users'** opinions about the organizations. **Targeted Twitter stream** is usually constructed by **filtering tweets** with user-defined selection criteria (*e.g.*, tweets published by users from a selected region, or tweets that match one or more predefined keywords). **Targeted Twitter stream** is then monitored to collect and **understand users'** opinions about the organizations. There is an emerging need for early crisis detection and response with such target stream. Such applications require a good **named entity recognition (NER)** system for *Twitter*, which is able to automatically discover emerging **named entities** that is potentially linked to the crisis. In this paper, we present a novel 2-step **unsupervised NER system** for targeted *Twitter* stream, called *TwNER*. In the **first step**, it leverages on the **global context** obtained from Wikipedia and **Web N-Gram corpus** to partition tweets into valid segments (phrases) using a **dynamic programming algorithm**. Each such tweet segment is a **candidate named entity**. It is observed that the **named entities** in the targeted stream usually exhibit a **gregarious property**, due to the way the targeted stream is constructed. In the **second step**, *TwNER* constructs a **random walk model** to exploit the **gregarious property** in the **local context** derived from the *Twitter* stream. The highly-ranked segments have a higher chance of being true **named entities**. We evaluated *TwNER* on **two sets of real-life tweets simulating two targeted streams**. Evaluated using labeled **ground truth**, *TwNER* achieves **comparable performance** as with conventional approaches in both streams. Various settings of *TwNER* have also been examined to verify our **global context + local context** combo idea.

TwNER: named entity
recognition in targeted
twitter stream
'SIGIR 2012

Classifier: Overview

Machine Learning algorithm:

Decision Trees from scikit-learn package.

Feature types:

- POS Tags and their derivatives
- External Knowledge Bases (DBLP, DBPedia)
- DBPedia relation graphs
- Syntactic features

Datasets

Two collections:

- CS Collection (SIGIR 2012 Research Track): 100 papers
- Physics collection: 100 papers randomly selected from [arXiv.org](https://arxiv.org) High Energy Physics category

	CS Collection	Physics Collection
N# Candidate N-grams	21 531	18 129
N# Judged N-grams	15 057	11 421
N# Valid Entities	8 145	5 747
N# Invalid N-grams	6 912	5 674

Available at: github.com/XI-lab/scientific_NER_dataset

Features: External Knowledge Bases

Domain-specific knowledge bases:

- DBLP (Computer Science): contains author-assigned keywords to the papers
- ScienceWISE: high-quality scientific concepts (mostly for Physics domain)

<http://sciencewise.info>



We perform exact string matching with these KBs.

ScienceWISE

Features: DBPedia, part I

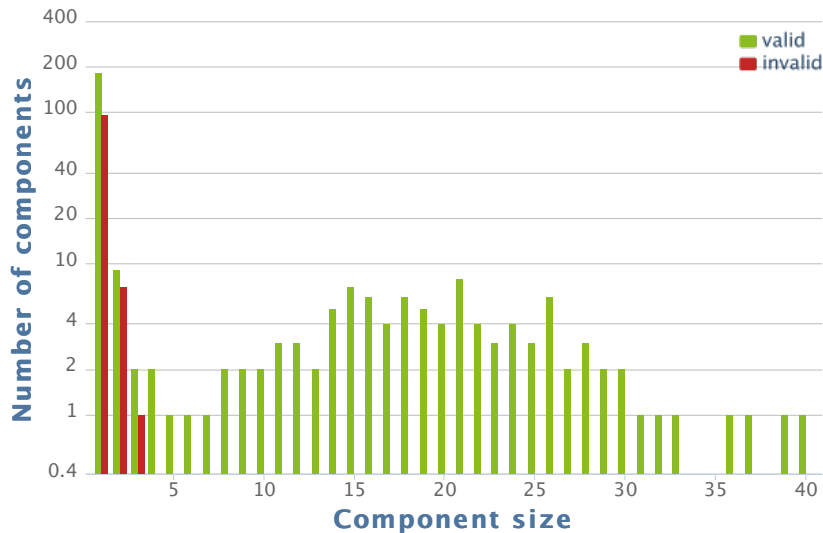
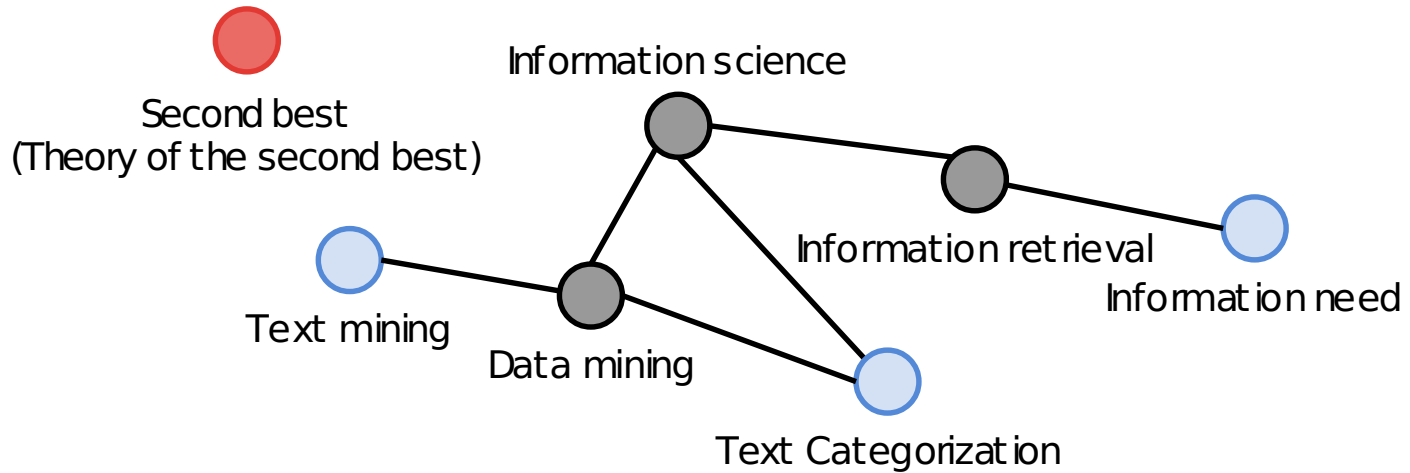
[DBPedia](#) pages essentially represent valid entities

But there are a few problems when:

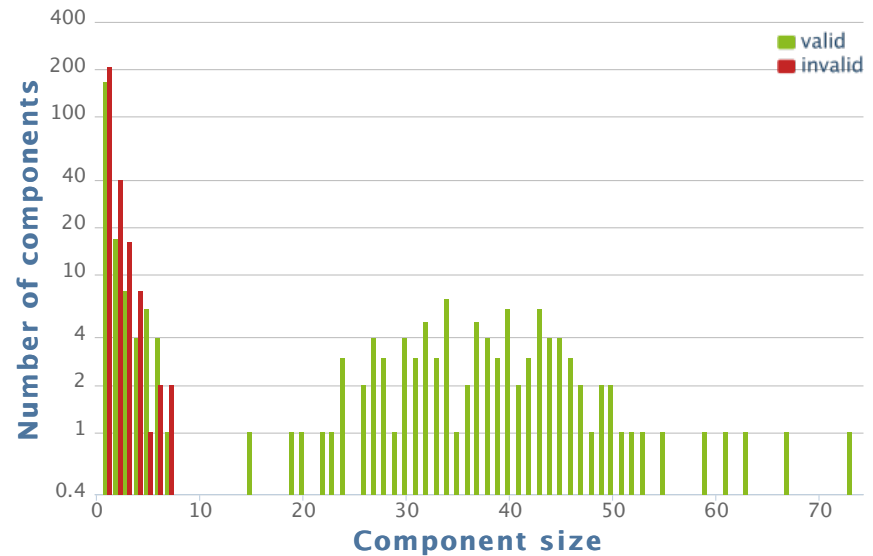
- N-gram is not an entity
- N-gram is not a scientific concept (“Tom Cruise” in IR paper)

	CS Collection		Physics Collection	
	Precision	Recall	Precision	Recall
Exact string matching	0.9045	0.2394	0.7063	0.0155
Matching with redirects	0.8457	0.4229	0.7768	0.5843

Features: DBPedia, part II



Without redirects



With redirects

Features: Syntactic

Set of common syntactic features:

- N-gram length **in words**
- Whether n-gram is uppercased
- The number of other n-gram given n-gram is part of

All results are obtained using 10-fold cross-validation.

Experiments: Feature Importance

	Importance
NN STARTS	0.3091
DBLP	0.1442
Components + DBLP	0.1125
Components	0.0789
VB ENDS	0.0386
NN ENDS	0.0380
JJ STARTS	0.0364

CS Collection, 7 features

	Importance
ScienceWISE	0.2870
Component + ScienceWISE	0.1948
Wikipedia redirect	0.1104
Components	0.1093
Wikilinks	0.0439
Participation count	0.0370

Physics Collection, 6 features

Experiments: MaxEntropy

MaxEnt classifier receives full text as input.
(we used a classifier from NLTK package)

Comparison experiment: 80% of CS
Collection as a training data, 20% as a test
dataset.

	Precision	Recall	F1 score
Maximum Entropy	0.6566	0.7196	0.6867
Decision Trees	0.8121	0.8742	0.8420

Lessons Learned

Classic NER approaches are not good enough for Idiosyncratic Web Collections

Leveraging the graph of scientific concepts is a key feature

Domain specific KBs and **POS patterns** work well

Experimental results show up to 85% accuracy over different scientific collections

Conclusions

- N-gram statistics for
 - Preposition correction
 - Named entity recognition for idiosyncratic documents
- Defined as binary classification problems
 - Over a set of features
- What works:
 - PMI, correlations, background knowledge bases/corpora