

Deliverable #4

Marie-Renée Arend

Josh Cason

Anthony Gentile

4 June 2013

Big idea: Classification

The screenshot shows the scikit-learn website homepage. At the top left is the scikit-learn logo. To its right are navigation links: Download, Support, User Guide, Examples, and Reference. Further right is a Google Custom Search bar. Below the navigation is a light blue banner with the text "scikit-learn: machine learning in Python". Underneath this banner are five visualization plots: 1. "Novelty Detection" showing a 2D scatter plot with decision boundaries and training/testing points. 2. "With connectivity constraints" showing a 3D ring-shaped data distribution. 3. "Bradypus variegatus" showing a map of South America with a color scale and AUC of 0.985. 4. "Microgonyx minutus" showing another map of South America with a color scale and AUC of 0.994. 5. "Non-negative components - NMF - Train time 0.8s" showing a grid of face images and their corresponding NMF components. On the left side of the page, there is an orange box with the text "The scikit-learn international code sprint is around the corner! Please, sponsor us" and a "News" section with the text "scikit-learn 0.13.1 is available for download. See what's new and tips on...". At the bottom of the page, there is a light orange banner with the text "Easy-to-use and general-purpose machine learning in Python" and "Scikit-learn integrates machine learning algorithms in the tightly-knit scientific Python world, building upon numpy, scipy...".

scikit-learn

Download Support User Guide Examples Reference

Google™ Custom Search Search

The **scikit-learn** international code **sprint** is around the corner! **Please, sponsor us**

News

scikit-learn 0.13.1 is available for download. See what's new and tips on

scikit-learn: machine learning in Python

Novelty Detection

With connectivity constraints

Bradypus variegatus

Microgonyx minutus

Non-negative components - NMF - Train time 0.8s

Easy-to-use and general-purpose machine learning in Python

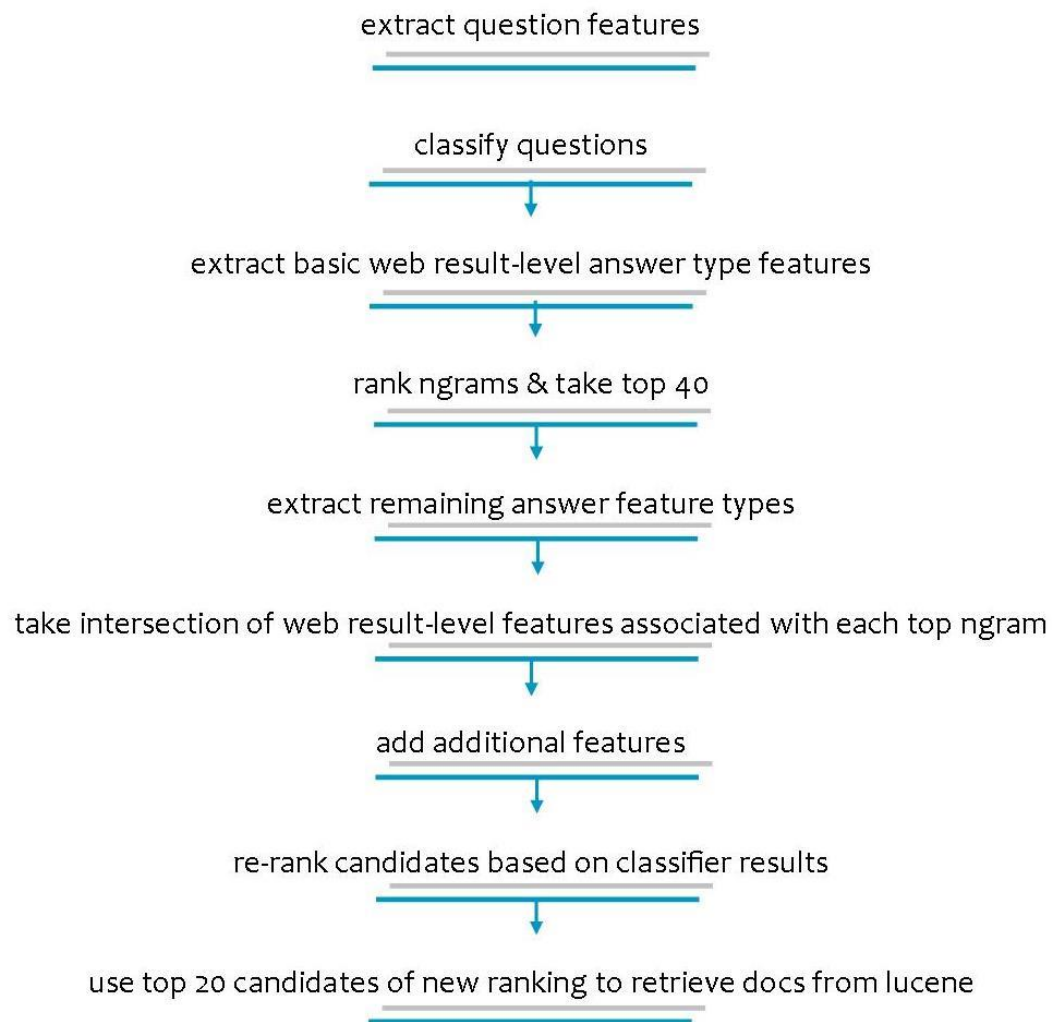
Scikit-learn integrates machine learning algorithms in the tightly-knit scientific Python world, building upon numpy, scipy...

- Scikit Learn python package
- Support Vector Machines classifier (Radial basis function kernel)
- Chi Squared feature selection

Big Idea: Caching

- Everything.

System Pipeline



Query Processing

- Approaches tried in previous versions:
 - D2: basic shallow processing
 - D3: using lexical resources
- Classifier approach:
 - D4: *loosely based on Li & Roth's syntactic features*
 - Stemmed ngrams ($n = 1,2,3,4$)
 - Weights for temporal, location or numerical question words
 - POS-tagged tokens from question & target with stopwords removed
 - Head NP & VP chunks – handwritten grammar
 - Question word(s)
 - *Issues:*
 - Addition of extra features beyond unigrams didn't make a significant difference & increased total runtime
 - Final system: features are unigrams

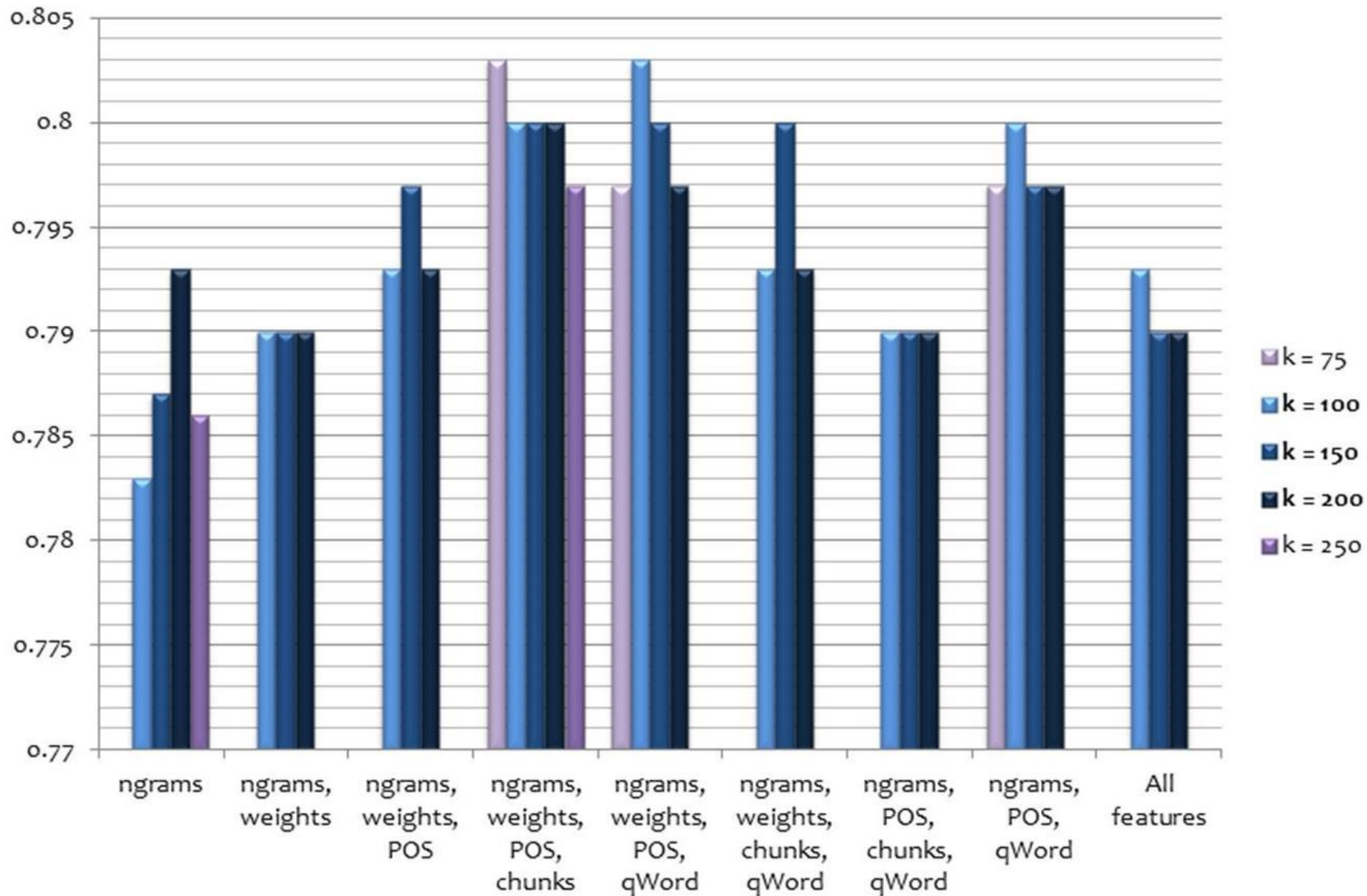


Fig. 1: Features and Performance (*experimentation phase*)

Classifier & Web-based Boosting

- Train question classifier (qc)
- Classify question
- Extract web result-level answer type features that require punctuation guided by qc
 - Before text processing a web result
 - take the qc, e.g., ABBR
 - extract all punctuation dependent ABBR patterns
 - `ABBR_PUNC_ABREV =`
'(M\.D\.|M\.A\.|M\.S\.|A\.D\.|B\.C\.|B\.S\.|Ph\.D|D\.C\.|NAAC
P|AARP|NASA|NATO|UNICEF|U\.S\.|USMC|USAF|USSR|Y
MCA)'

Classifier & Web-based Boosting

- Tokenize, remove punct., etc
- Re-rank ngrams & take top 40
 - Use Lin's web redundancy algorithm for re-ranking
- Extract ngram level answer pattern features as guided by qc
 - Similar to above but based on a particular answer candidate – no punctuation patterns
 - *(more info below)*

Classifier & Web-based Boosting

- Add the intersection of all web result-level features associated with each top-40 ngram, n
 - $\bigcap_{w \in W} f(n, w)$
 - Where f returns the set of features for w if n appeared there
- Add additional features like top web result rank

Classifier & Web-based Boosting

- Re-rank based on classifier
 - Each candidate is assigned a probability of being a “yes” answer
 - Training based on checking 2004, 2005 answer candidates against their answer patterns using same features
- Use the top 20 candidates from the new ranking to retrieve docs using lucene

Answer Pattern Detection

We used a set of regular expressions to detect answer types in addition to our existing filters and weighting logic.

If we have a question classified as type:

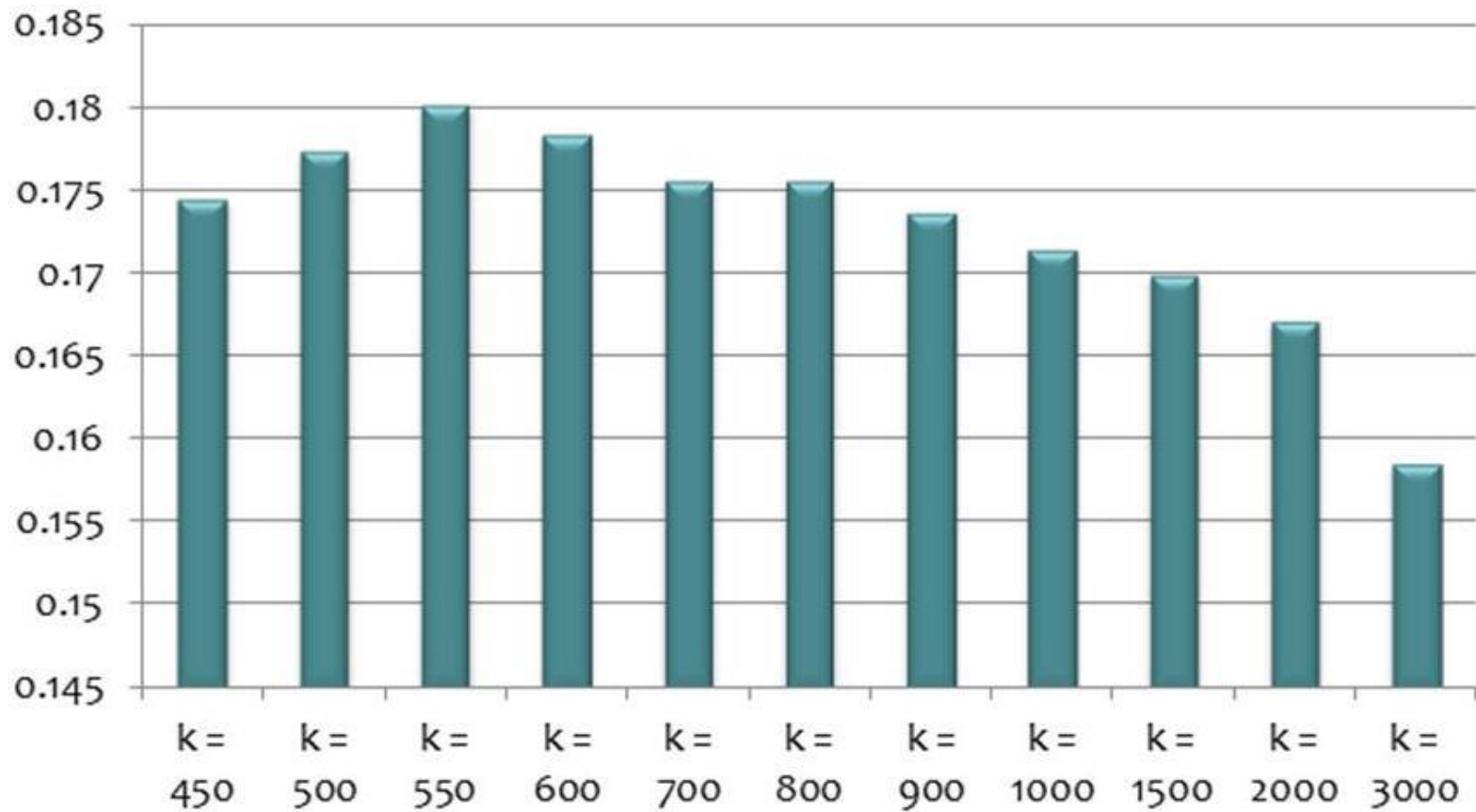
```
['LOC', 'HUM', 'NUM', 'ABBR', 'ENTY', 'DESC']
```

If 'ENTY', a set of regular expressions for subclasses are triggered (sports, religion, colors, etc):

Example:

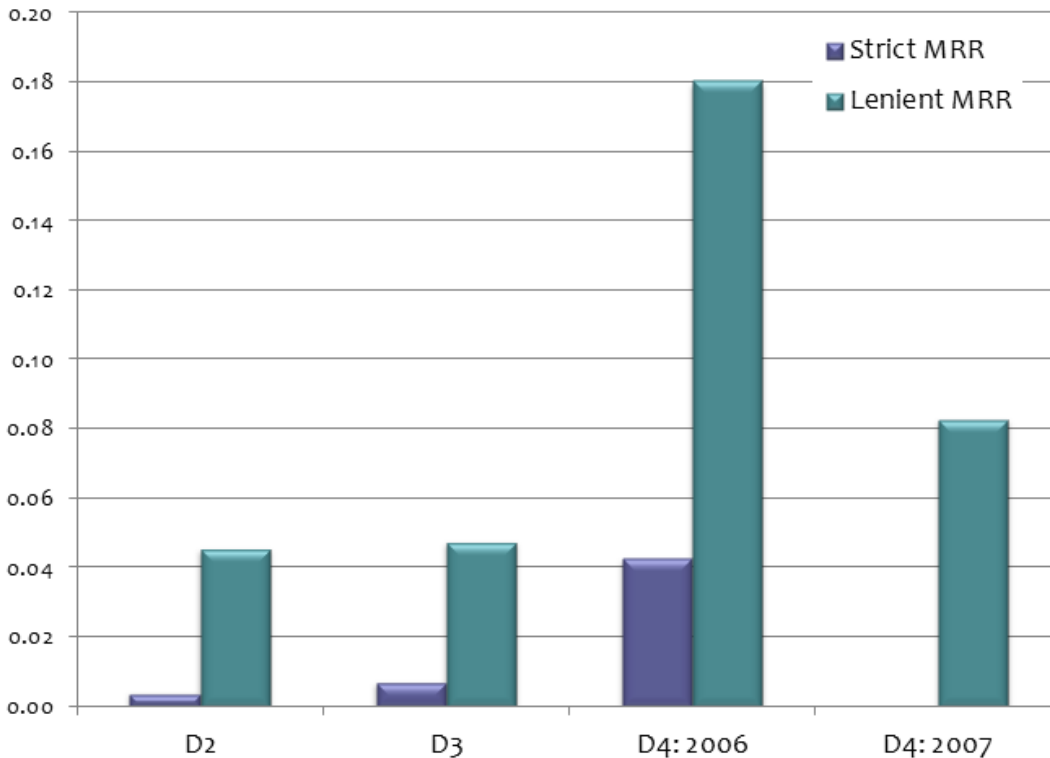
```
ENTY_PLANTS =  
set(['rose', 'weed', 'tulip', 'daisy', 'flower', 'orchid', 'bonsai', 'dog  
wood'])  
pattern_values['plant'] = ['(' + '|'.join(self.ENTY_PLANTS) + ')']
```

This pattern dictionary is iterated over to find matches in the text and provide for features and boost in weighting for the web results.



Experiment: Select k best features using X^2 selection
(Numbers are lenient MRR scores for 2006)

Results, Issues & Successes



- Results analysis
- Issues
 - 0 for 2007 strict MRR
- Successes
- Notes:
 - All answer candidates were less than or equal to 100 chars

	Strict MRR	Lenient MRR
D2	0.0039	0.0455
D3	0.0070	0.0475
D4: 2006	0.0426	0.1801
D4: 2007	0.0000	0.0828

Resources

- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.
- Graff, D. (Ed.). (2002). *The AQUAINT corpus of English news text*. Linguistic Data Consortium.
- Hatcher, E., Gospodnetic, O., & McCandless, M. (2004). *Lucene in action*.
- Li, X. & Roth, D. (2005). Learning question classifiers: The role of semantic information. *Natural Language Engineering*, 1(1), Retrieved from <http://12.cs.uiuc.edu>
- Lin, J. (2007). An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems (TOIS)*, 25(2), 6.
- Mishne, G. & de Rijke, M. (2005). *Query formulation for answer processing*. Published research, Informatics Institute, University of Amsterdam. Retrieved from <http://dare.uva.nl>
- Resnik, Philip. (1995). Disambiguating Noun Groupings with Respect to WordNet Senses. *Third Workshop on Very Large Corpora*. Retrieved from <http://acl.ldc.upenn.edu/W/W95/W95-0105.pdf>