

# Entity Set Refinement & Semantic Labeling of Entity Sets

---

Günter Neumann, LT-lab, DFKI, December 2011

# Vishnu Vyas & Patrick Pantel: Semi-Automatic Entity Set Refinement, ACL-2009

---

- Starting point: Compute set expansion ala Sarmento et al., 2007
- Observation: Since similarity is computed between all pairs, it is possible to identify similarities between false positives FP
- Approach for handling FP: **user relevance feedback**
- Core idea: after each iteration: manually check expanded set, and for one **identified error X**, **delete** all Y, which are **highly similar** to X

# Identifying errors

---

- Core idea is to **improve the precision** of a set expansion method by **using minimal human negative judgements**.
  - Leverage the fact that errors are systematically caused by **ambiguous seed instances** which attract incorrect instances of an unintended entity type.
  - For error identification, use **distributional similarity** to identify relevant features which can be used to quickly clean up an expanded set.
- The algorithm is a **post-processor for set expansion methods** using minimal human feedback.
  - **relevance feedback** in form of negative examples

# Ambiguous Seeds: Two Observations

---

- a) Many expansion errors are systematically caused by **ambiguous seed** examples.
- Example: **Neptune** ? a roman god or a planet name
  - Problem: such an example can cause identification of wrong set members
- b) Entities which are similar in one sense are usually not similar in their other senses (**ako of restricted type-preserving compositionality**).
- Example: **Apple** and **Sun** have **similar company sense**, but their other senses (**Fruit, Celestial Body**) are not similar.

# Two Methods for Identifying and Correcting Set Expansion Errors

---

- SIM: Similarity Method
- FMM: Feature Modification Method
- Approximate Cosine Similarity

# SIM: Similarity Method for Handling Case a)

- Compute **feature vector**  $\mathbf{x} \in \mathbf{V}$  for each term of the expanded set

- Record surrounding context of **each term**

- In this paper:

- window size of 1 (so left/right word used as context element for a term)

- PMI (point-wise mutual information) to weight contexts

- cosine score for similarity, i.e.,  $\cos(x,y)$  for  $x, y \in V$

- APSS (all pairs similarity search) used for computing similarity between all possible term pairs (i.e., pairs of feature vectors)

- **Similarity matrix directly used to refine entity sets:** Given a manually identified error  $x \in V$ , in each iteration, automatically remove all elements that are **semantically similar to x** (i., find all pairs  $(x,y)$  s.t.  $y \in V$  &  $\cos(x,y) \geq t$ , where  $t$  is an experimentally determined threshold)

**Pointwise mutual information**

This is a method discussed in *Social Media Analysis 10-802* in Spring 2010.

If  $X$  and  $Y$  are random variables, the pointwise mutual information between two possible outcomes  $X=x$  and  $Y=y$  is

$$PMI(x,y) = \log \frac{Pr(X=x, Y=y)}{Pr(X=x)Pr(Y=y)}$$

This quantity is zero if  $x$  and  $y$  are independent, positive if they are positively correlated, and negative if they are negatively correlated.

In *Turney, ACL 2002* this was used as a way of assessing the *semantic orientation* of words or phrases. Specifically the semantic orientation of  $x$  was defined as

$$SO(x) = PMI(x, \text{'excellent'}) - PMI(x, \text{'poor'})$$

In more detail, Turney interpreted " $X=x$  and  $Y=y$ " as an event where two words  $x$  and  $y$  occur nearby in the same document, and " $X=x$ " as an event where word  $x$  occurs in a document. After some simplification,  $SO(x)$  can then be written as

$$\log \frac{Hits(x \text{ near 'excellent'}) \cdot Hits(\text{'poor'})}{Hits(x \text{ near 'poor'}) \cdot Hits(\text{'excellent'})}$$

This means that  $SO(x)$  can be computed quickly - with just two queries to a search engine.

# FMM: Feature Modification Method for Handling Case b)

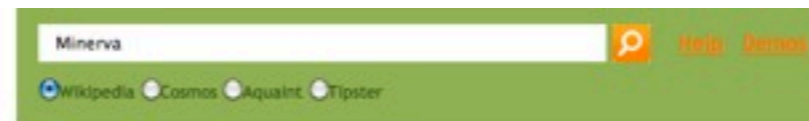
---

- Automatically **discover the incorrect contexts** of the unintended senses of seed elements
- Filter out expanded terms whose contexts do not overlap with the other contexts of the seed elements.
- Difference to SIM:
  - In SIM all elements which have a feature vector similar to the error  $e$  are removed.
  - **FMM tries to identify the subset of features** of error  $e$  that represent the unintended sense of the seed set  $S$ .

# FMM: Modify Centroid of Seed Elements

- Construct a **centroid vector C** for S by taking a **weighted average** of the contexts of the **seed elements** in S
- Can be obtained from a large corpus s.t. Wikipedia
- Use centroid vector C for identifying „wrong senses“

<http://demo.patrickpantel.com/demos/lexsem/features.htm>



## Minerva

-V:subj:N 216 times:

attack 6, destroy 6, head 4, pursue 4, return to 4, fire 3, launch 3, participate 3, star 3, ash 2, bear 2, call 2, come 2, engage in 2, explode 2, fall 2, get into 2, investigate 2, kill 2, learn 2, retransform 2, abduct 1, accept 1, access 1, aid 1, announce 1, bathe 1, Beat 1, betray 1, blame 1, breakthrough 1, break up 1, bring about 1, catch 1, collapse 1, complete 1, compliment 1, c 1, discard 1, get involved in 1, hurt 1, hurt 1, insult 1, intend 1, interrogate 1, land 1, last 1, overthrow 1, pinpoint 1, proceed 1, race 1, BAM 1, resemble 1, Sabotage 1, Sigh 1, slip away 1, talk about 1, transmit 1, wait 1

N:det:Det 250 times:

the 244

-N:nn:N 144 times:

x 15, pic 6, pilot 4, series 4, Captain 3, cast 3, engine 3, reef 3, CORVETTE 2, Edition 2, flume 2, group 2, med 2, Nine 2, plan 2, religion 2, studio 2, I 1, J 1, @ 1, arrival 1, BC 1, bear 1, Chichester 1, college 1, console 1, convent 1, Cowboy Bebop 1, dealer 1, Debbie Reynolds 1, design 1, designer 1, environment 1, executive officer 1, factory 1, field 1, flag 1, fraternity 1, ft Journal 1, last 1, line 1, logo 1, magazine 1, medal 1, milk 1, M 1, Monteverdi 1, Mouse 1, name 1, plot 1, pursuit 1, repair 1, Rose 1, slaughter 1, Smith 1, tortle 1, source code 1, statue 1, J 1, theater 1, virtue 1, visit 1, weaponry 1

N:nn:N 270 times:

sopra 66, Santa Maria 59, Goddess 16, battleship 11, della 10, piazza 7, ZAFI 7, S. Maria 6, ship 5, di 4, Roman 3, wife 3, @@@@-@@@@ 2, Athena 2, Ida 2, ISSN 2, planet 2, robot 2, There 2, baroque 1, Basilica 1, beam 1, church 1, Congress 1, corns 1, CORVETTE 1, Deftones 1, enclave 1, fact 1, flagship 1, follow-up 1, Galleria 1, gvoid 1, leaflet 1, libertarian 1, M. 1, magazine 1, Otto 1, player 1, Poland 1, sealer 1, Stalinist 1, study 1, Temple 1, whaling ship 1, BC 1

-N:conj:N 123 times:

Juno 24, Diana 5, Athena 4, Jupiter 4, Astarte 3, Lucia 3, Mars 3, Neptun 3, Venus 3, Archangel 2, Dexter 2, left 2, Perceptor 2, portrait 2, William 2, agreement 1, Alpheus 1, alternative 1, asteroid 1, Athrun 1, battleship 1, cab 1, Cupid 1, Finance 1, Florence 1, Flute 1, foot 1, forge 1, freighter 1, Glady 1, Gondwana 1, habit 1, Hercules 1, himself 1, Journal 1, Mercury 1, Nile 1, Rolls-Royce 1, Shin 1, sister 1, Space 1, statue 1, their 1, Viktoria 1, Vulcan 1

N:conj:N 69 times:

Goddess 3, Urania 3, father 2, Jupiter 2, prudence 2, Wheeljack 2, @@@@-@@@@ 1, Anna Maria 1, Apollo 1, Archangel 1, Artemis 1, Aurora 1, Captain 1, Cerns 1, Chester 1, Daimier 1, de engine 1, Eursoa 1, fight 1, flume 1, fortress 1, Hercules 1, Mars 1, Mercury 1, Mus 1, mmsh 1, patria 1, pellet 1, portraval 1, Rome 1, scheme 1, sister 1, theatre 1, those 1, young man 1

-N:appo:N 33 times:

@@@@ 2, battleship 2, sister 2, Carnival 1, doll 1, grandmother 1, great-granddaughter 1, gun 1, John 1, Juno 1, model 1, palace 1, paperback 1, predecessor 1, Rome 1, ship 1, single 1, theati

N:appo:N 38 times:

Goddess 6, G. 2, @@@@-@@@@ 1, @@@ 1, acropolis 1, decq 1, Francis 1, gun 1, June @@@@ 1, Juno 1, lbs 1, March @@@ 1, p.62 1, protector 1, psyche 1, remains 1, Rome 1, sculpture 1, sliv 1

Minerva 14 elements



# FMM: Example - Roman Gods

---

- $S = \{\text{Minerva, Neptune, Bacchus, Juno, Apollo}\}$ ,
- features of centroid  $C = \text{attach, kill, *planet, destroy, Goddess, *observe, statue, *launch, Rome, *orbit}$
- let new wrong element be  $e = \text{Earth}$  (planet sense):
- then **removing the intersecting** features from  $e$  and  $C$  will remove the „planet sense“ features (marked with \*) caused by the **seed element Neptune**
- **then** compute  $\cos(C_{\text{modified}}, y)$

# FMM: Efficiency - Approximate Cosine Similarity

---

- FMM requires online similarity computations between (modified) centroid vectors and all elements of the expanded set (to find and remove those elements which are similar to manually selected error).
- Problem: feature vector can be in the GB for large corpora; thus storing the feature vectors for all candidate expansions and the seed set is inefficient, especially for an interactive system
- Idea: only store the **shared features between the centroid and the words** rather than the complete feature space
- Note: features (i.e., contexts) are only removed from the original centroid - no new features are ever added.

# Approximate Cosine Similarity - Idea

---

- Let  $y$  be the original centroid representing  $S$  and expansion error  $e$ . Let  $y'$  be the modified centroid by removing the features of  $e$  from  $y$ .
- FMM requires computing the similarity between all expanded elements  $x$  and  $y'$  (i iterates over feature space).

$$\cos(x, y') = \frac{\sum x_i y'_i}{\|x\| * \|y'\|}$$

- The norm of  $x$  is the only element that is not known. Assuming we know the original similarity  $\cos(x, y)$  & the shared elements

$$\|x\| = \frac{\sum x_i y_i}{\cos(x, y) * \|y\|}$$

- Combining both gives

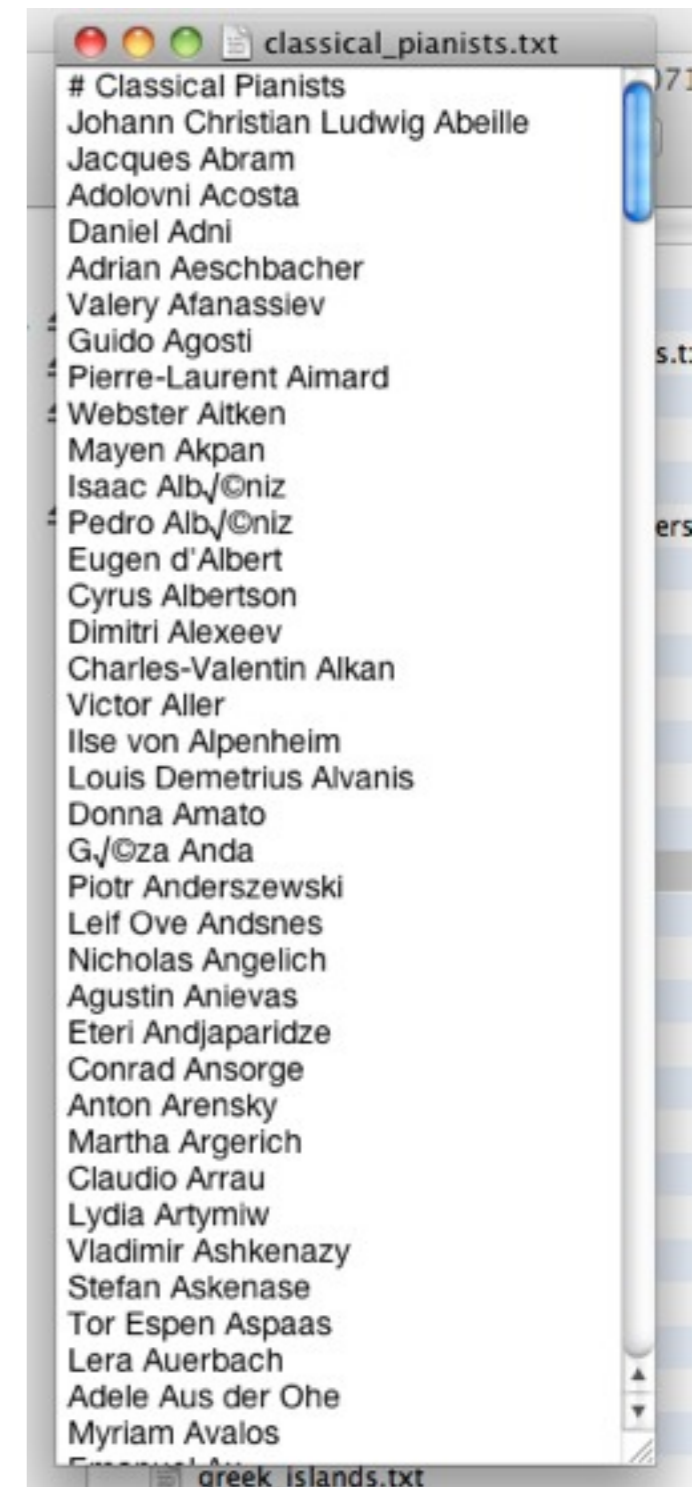
- can be considered as an update of the original cosine score where the update only depends on the shared features and the original centroid.

$$\cos(x, y') = \cos(x, y) * \frac{\sum x_i y'_i}{\sum x_i y_i} * \frac{\|y\|}{\|y'\|}$$

# Datasets and Baseline

- Manual selected gold standard sets, which were extracted from Wikipedia
- Selected from **List of** Wikipedia pages:
  - Collect all nouns from Wikipedia; sort them randomly
  - For each noun, check whether it exists in a Wikipedia list (a max. of 50 lists are considered)
  - If so, extract that list, and if noun exists in different lists, manual selection of best list by authors
  - The 50 lists contained 208 elements on average (from 11 to 1116).
  - Data set: /Users/gunterneumann/dfki/data/wikipedia.20071218.goldsets, e.g., „classical pianists“
- More than 1000 trial sets are generated automatically by randomly sorting and selection of max. 20 seed elements

Note: wikipedia XML files can be [download from here](#)



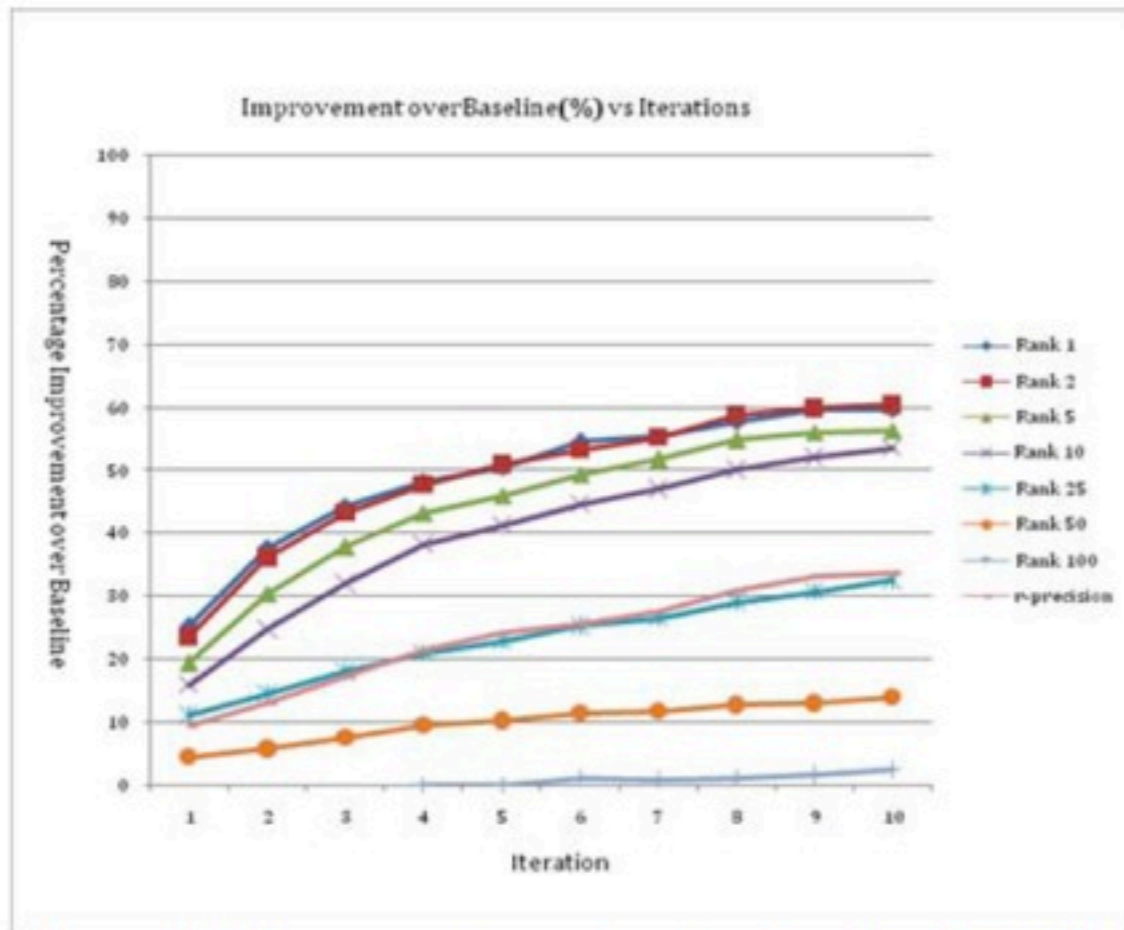
```
classical_pianists.txt
# Classical Pianists
Johann Christian Ludwig Abelle
Jacques Abram
Adolovni Acosta
Daniel Adni
Adrian Aeschbacher
Valery Afanassiev
Guido Agosti
Pierre-Laurent Aimard
Webster Aitken
Mayen Akpan
Isaac Albizniz
Pedro Albizniz
Eugen d'Albert
Cyrus Albertson
Dimitri Alexeev
Charles-Valentin Alkan
Victor Aller
Ilse von Alpenheim
Louis Demetrius Alvanis
Donna Amato
Gyza Anda
Piotr Anderszewski
Leif Ove Andsnes
Nicholas Angelich
Agustin Anievas
Eteri Andjaparidze
Conrad Ansorge
Anton Arensky
Martha Argerich
Claudio Arrau
Lydia Artymiw
Vladimir Ashkenazy
Stefan Askenase
Tor Espen Aspaas
Lera Auerbach
Adele Aus der Ohe
Myriam Avalos
```

# Simulation of User Feedback & Baseline

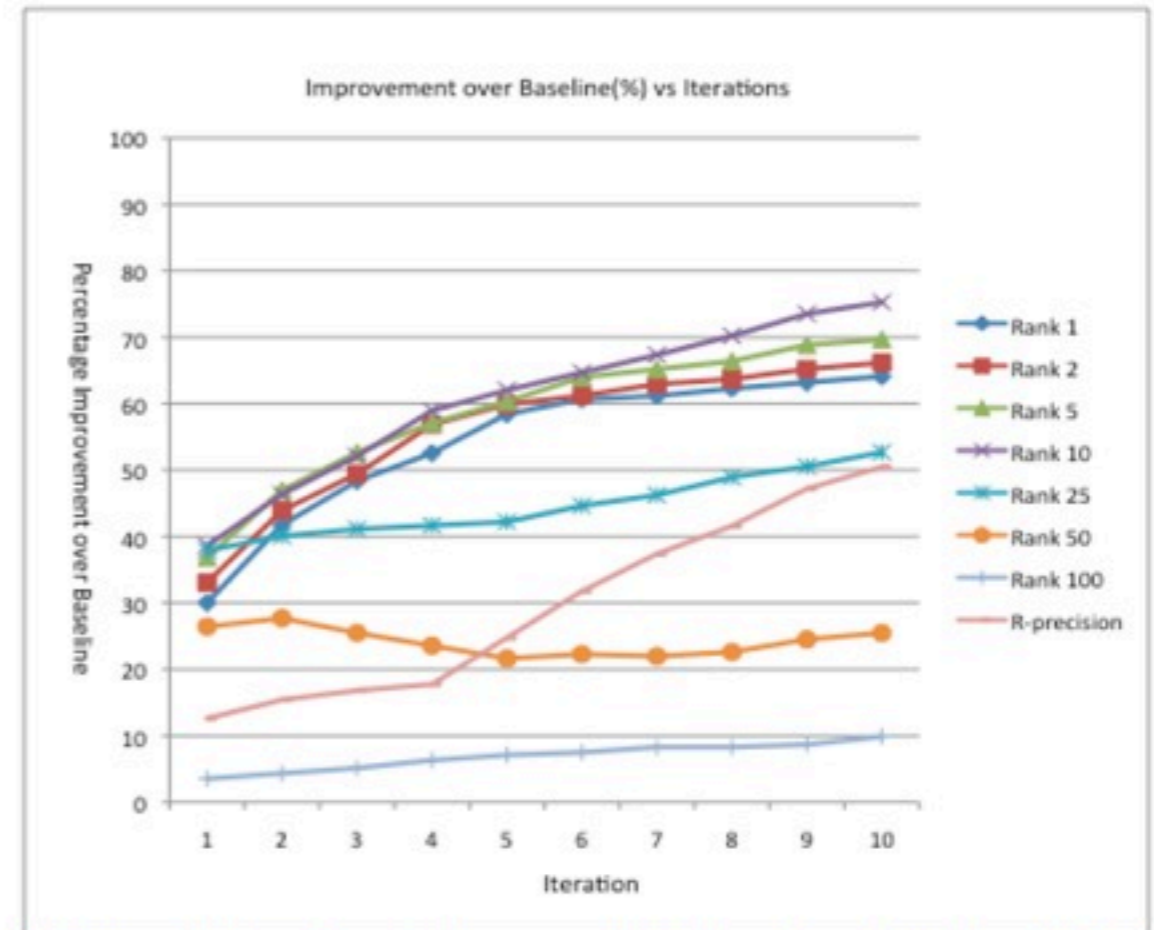
---

- Use of gold standard sets as judgement of expansion candidates
- In each iteration, the first expansion that was judged incorrect was used as negative example
- Baseline method:
  - measure improvement by **just removing the first incorrect entry** (as above), i.e., no automatic identification of similar erroneous elements
  - this simulates the process of manually cleaning a set by removing one error at a time

# Results



**Figure 1.** Precision gain over baseline algorithm for SIM method.



**Figure 2.** Precision gain over baseline algorithm for FMM method.

# Results

- Wikipedia used as source corpus
  - POS-tagged and chunked
- Corpus statistics for SIM and FMM
  - computed over the semi-syntactic contexts (chunks)
  - minimum similarity thresholds are set to SIM=0.15 and FMM=0.11 (used for identifying similar negative examples)
- For each trial set: Set expansion via Sarmiento et al. 2007
  - Then judgement against gold standard, where each candidate was marked as either correct or wrong.
- These expanded trial sets are then inputted to SIM and FMM

**Table 1.** R-precision of the three methods with 95% confidence bounds.

<i>ITERATION</i>	<i>BASILINE</i>	<i>SIM</i>	<i>FMM</i>
1	0.219±0.012	<b>0.234±0.013</b>	0.220±0.015
2	0.223±0.013	<b>0.242±0.014</b>	0.227±0.017
3	0.227±0.013	<b>0.251±0.015</b>	0.235±0.019
4	0.232±0.013	<b>0.26±0.016</b>	0.252±0.021
5	0.235±0.014	<b>0.266±0.017</b>	<b>0.267±0.022</b>
6	0.236±0.014	0.269±0.017	<b>0.282±0.023</b>
7	0.238±0.014	0.273±0.018	<b>0.294±0.023</b>
8	0.24±0.014	0.28±0.018	<b>0.303±0.024</b>
9	0.242±0.014	0.285±0.018	<b>0.315±0.025</b>
10	0.243±0.014	0.286±0.018	<b>0.322±0.025</b>

FMM: 32.5% improvement after 10 iterations:  
 $(322-243)*100 / 243$

R-precision: precision at the size of the gold standard set; e.g., if GS set contains 20 elements, then R-precision for any set expansion is measured as the precision at rank 20 (20 best candidates); the average R-precision over each trial set is reported

Van Durme & Pasca, AAAI-2008

## Finding Cars, Goddesses and Enzymes: Parametrizable Acquisition of Labeled Instances for Open-Domain Information Extraction

---

- Contribution:
  - A TFIDF like method is developed for deriving labeled classes of instances from unstructured, open domain text.
  - **Filtering of is-a extraction pairs** through distributionally similar terms.
  - Results depend on number of extracted classes: 440 classes = 91% accuracy, 8,572 classes = 86%.
- Work builds on top of Pantel & Ravichandran, 2004 „Automatically Labeling Semantic Classes. HLT-NAACL.“ (They already provide the basis for reasonably class labeling, but still imperfect).



# Motivation

---

- Goal: automatically construct large knowledge bases from text.
- Web-scale extraction: recent approaches often manually specify the type of knowledge to be extracted in advance, e.g., through small sets of seed elements
  - Progress is being hampered by the lack of a reliable resource containing a diverse set of classes represented through sets of representative instances.
  - Since such a resource does not exist, either user coarse-grained classes (e.g., person, location, ...) or create new, experimental classes manually.
- Limitation: scale of experimentation and diversity of class members.
- **Solution: Unsupervised clustering and automatic labeling of identified classes.**

# Extraction Algorithm: given

---

- a large collection of **is-a extraction pairs**  $P := \{ \langle I, L \rangle \mid I \text{ an instance, } L \text{ a label of } I \}$ , e.g.,  $\langle \text{Barack Obama, president} \rangle$ 
  - pairs might be extracted from ISA extraction patterns following Hearst (e.g., „x is a y“, „y such as x“)
- **clusters** of semantically related instances,  $C$ , s. a.  $\{ \text{George Bush, Bill Clinton, Barack Obama} \}$ 
  - distributional similarity:  
„Clinton vetoed the Bill.“ & „Bush vetoed the bill“  $\rightarrow$  Clinton and Bush are semantically similar
  - clusters are a partitioning of  $I$ , computed for example via distributional similarity following Lin & Pantel, 2002 (Concept Discovery from Text via Clustering by committee - CBC):
    - two instances are similar, if they tend to appear in similar context (distributional hypothesis);
    - compute centroid of each cluster, s.t. the centroid of a cluster is constructed by averaging the feature vectors (of the instances) of a (carefully selected) subset of the cluster members. The subset is viewed as a committee that determines which other elements belong to the cluster.

# CBC - Example Clusters

---

(A) multiple sclerosis, diabetes, osteoporosis, cardiovascular disease, Parkinson's, rheumatoid arthritis, heart disease, asthma, cancer, hypertension, lupus, high blood pressure, arthritis, emphysema, epilepsy, cystic fibrosis, leukemia, hemophilia, Alzheimer, myeloma, glaucoma, schizophrenia, ...

(B) Mike Richter, Tommy Salo, John Vanbiesbrouck, Curtis Joseph, Chris Osgood, Steve Shields, Tom Barrasso, Guy Hebert, Arturs Irbe, Byron Dafoe, Patrick Roy, Bill Ranford, Ed Belfour, Grant Fuhr, Dominik Hasek, Martin Brodeur, Mike Vernon, Ron Tugnutt, Sean Burke, Zach Thornton, Jocelyn Thibault, Kevin Hartman, Felix Potvin, ...

(C) pink, red, turquoise, blue, purple, green, yellow, beige, orange, taupe, white, lavender, fuchsia, brown, gray, black, mauve, royal blue, violet, chartreuse, teal, gold, burgundy, lilac, crimson, garnet, coral, grey, silver, olive green, cobalt blue, scarlet, tan, amber, ...

# CBC - Example Clusters

---

**(A)** multiple sclerosis, diabetes, osteoporosis, cardiovascular disease, Parkinson's, rheumatoid arthritis, heart disease, asthma, cancer, hypertension, lupus, high blood pressure, arthritis, emphysema, epilepsy, cystic fibrosis, leukemia, hemophilia, Alzheimer, myeloma, glaucoma, schizophrenia, ...

**(B)** Mike Richter, Tommy Salo, John Vanbiesbrouck, Curtis Joseph, Chris Osgood, Steve Shields, Tom Barrasso, Guy Hebert, Arturs Irbe, Byron Dufoe, Patrick Roy, Bill Ranford, Ed Belfour, Grant Fuhr, Dominik Hasek, Martin Brodeur, Mike Vernon, Ron Tugnutt, Sean Burke, Zach Thornton, Jocelyn Thibault, Kevin Hartman, Felix Potvin, ...

**(C)** pink, red, turquoise, blue, purple, green, yellow, beige, orange, taupe, white, lavender, fuchsia, brown, gray, black, mauve, royal blue, violet, chartreuse, teal, gold, burgundy, lilac, crimson, garnet, coral, grey, silver, olive green, cobalt blue, scarlet, tan, amber, ...

- Limitation: does not discover names for the clusters !
- Would be nice to have class names, e.g., in case of Question Answering Systems. E.g., Given class (B) and a label like „goaltender“ as expected answer type
- a QA system could answer a question like „Which goaltender won the most Hart Trophys?“

# Core Idea of a Semantic Labeling Approach

---

- Compute clusters  $C$ .
- For each instance in a cluster  $c \in C$  find candidate IS-A pairs using Hearst patterns.
- Perform some voting strategy to identify the most plausible label from all labels determined for the instances of class  $c$ .
- Can be improved by selecting a subset of instances of class  $c$  - called the „committee“ of  $c$  - from which label candidates are computed

1) cardiovascular disease, diabetes,  
multiple sclerosis, osteoporosis,  
Parkinson's, rheumatoid arthritis

2) Curtis Joseph, John Vanbiesbrouck, Mike  
Richter, Tommy Salo

3) blue, pink, red, yellow

# Extraction Algorithm: TFIDF-based strategy

**Given:**  $\mathcal{I}$  : set of instance phrases  
 $\mathcal{L}$  : set of label phrases  
 $\mathcal{C}$  : partitioning of  $\mathcal{I}$  by distributional similarity  
 $\mathcal{P} \subseteq \mathcal{I} \times \mathcal{L}$  : set of *is-a* phrase pairs

**Returns:**  $\mathcal{P}_{JK} \subseteq \mathcal{P}$  : set of filtered phrase pairs

**Parameters:**  $J \in [0, 1]$  : label freq. constraint (intra-cluster)  
 $K \in \mathbb{N}$  : label freq. constraint (inter-cluster)

**Algorithm:**  
 Let  $\mathcal{P}_{JK} = \{\}$   
 For each semantic cluster  $S \in \mathcal{C}$  :  
   For each class label  $L$ , where  $\exists I \in S$  s.t.  $\langle I, L \rangle \in \mathcal{P}$  :  
     Let  $S_L = \{I \mid I \in S, \langle I, L \rangle \in \mathcal{P}\}$   
     Let  $\mathcal{C}_L = \{S' \mid S' \in \mathcal{C}, \exists I \in S' : \langle I, L \rangle \in \mathcal{P}\}$   
     If  $|S_L| > J \times |S|$  :  
       If  $|\mathcal{C}_L| < K$  :  
         Set  $\mathcal{P}_{JK} = \mathcal{P}_{JK} \cup \{\langle I, L \rangle \mid I \in S, \langle I, L \rangle \in \mathcal{P}\}$

Figure 1: Algorithm for extracting  $\langle$ instance, class label $\rangle$  pairs.

## TFIDF interpretation:

TF is the number of instances in a cluster initially assigned a given label  $L$ , divided by the total number of instances  $I$  in that cluster.

IDF is used based on the belief that labels spread over many clusters are less significant.

- Goal of filtering method: filter out erroneous pairs
- $S_L :=$  all instances with same label from cluster  $S$
- $\mathcal{C}_L :=$  all other clusters  $S'$  which have at least one instance labeled with  $L$
- Check whether  $S_L$  is viable:  $|S_L|/|S| > J$  ( $J = 0 \dots 1$ ) (TF for label  $L$ )
- Then if  $|\mathcal{C}_L| < K$  ( $K$  in  $\mathbb{N}$ ) then  $L$  is considered a good label for supporting instances in  $S$  (i.e., the elements  $I$  in  $S$  with label  $L$ ) (IDF for label  $L$ )

# Experimental Setting

---

Possible to use uniform  
tagger to achieve multi-  
linguality.  
GN: test this !

- Unstructured text from **100 million Web documents** (from Google snap shot)
- Documents are tokenized and **POS-ed** stream of sentences
- Clusters of related terms ala Lin & Pantel, 2002
- Initial instance label pairs via isa patterns ala Hearst, 1992 (e.g., „X such as Y  
„ or „X including Y“)
- WN400K: automatically expanded 400.000+ WordNet synsets; used as additional background knowledge base during evaluation

# Parameter Setting

- Use  $J$  and  $K$  as parameters:
  - $J :=$  effects the **number of instances within a class** that must share a label for it to be viable
  - $K :=$  effects the **distribution across clusters** a label is allowed to be
- Different settings:
  - if  $J$  is lowered then increase in the number of resultant classes (the smaller the size of elements is with same  $L$  in a particular cluster)
  - if  $K$  is increased then the more distributed across clusters a label is allowed to be (the more ambiguous label  $L$  is)

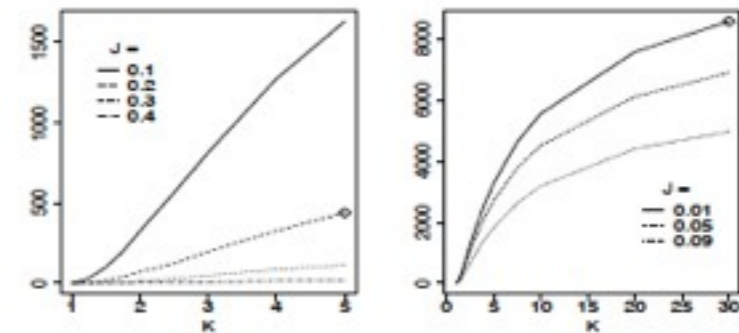


Figure 2: Number of classes extracted at strict, and less prohibitive settings of  $J$  and  $K$ .

size	number	size	number
$\leq \infty$	8,572	$\leq 25$	4,322
$\leq 500$	8,311	$\leq 10$	1,681
$\leq 50$	6,089	$\leq 5$	438

Table 1: For  $J = 0.01$ ,  $K = 30$ , number of classes whose size  $\leq N$ .

Settings in the experiments:  
 wide ( $J=0.01$ ,  $K=30$ )  
 narrow ( $J=0.2$ ,  $K=5$ )

USILOM ( $\eta=0.5$ ,  $K=2$ )



# Evaluation: Instances

---

- Instance vocabulary evaluation
  - 100 randomly selected instances
- Table 3: near perfect result (when ignoring label)
- Judgment by authors of paper
- Precision > 97%

<i>J</i>	<i>K</i>	Good	%
0.01	30	97/100	97 $\pm$ 3.3%
0.2	5	98/100	98 $\pm$ 2.7%

Table 3: Assessed quality of underlying instances.

Instance	Good?	Instance	Good?
<i>fast heart rate</i>	yes	<i>electric bulb</i>	yes
<i>local produce</i>	yes	<i>south east</i>	yes
<i>severe itching</i>	yes	<i>finding directions</i>	yes
<i>moles and voles</i>	no	<i>h power</i>	no

Table 4: Select examples from instance assessment.

# Evaluation: Class Labels

- 100 randomly selected pairs for wide and narrow settings
- Baseline: similar sample was assessed taken directly from the input (i.e., without the filtering)
- Checking novelty: removing all pairs that are contained in WN400K
- Table 5: significant is the difference between wide/narrow settings
- Precision is > 91%

$J$	$K$	$\mathcal{P}_{JK}$		$\mathcal{P}_{JK} \setminus \{\langle \mathcal{I}_{400k}, \cdot \rangle\}$	
		Eval	Precision	Eval	Precision
0	$\infty$	34/100	34 $\pm$ 9.3%	27/100	27 $\pm$ 8.1%
0.01	30	86/100	86 $\pm$ 6.9%	75/100	75 $\pm$ 8.5%
0.2	5	91/100	91 $\pm$ 5.6%	95/100	95 $\pm$ 4.3%

Table 5: Quality of pairs, before and after removing instances already in WN400k.

Instance	Class	Good?
<i>go-karting</i>	<i>outdoor activities</i>	yes
<i>ian and sylvia</i>	<i>performers</i>	yes
<i>italian foods</i>	<i>foods</i>	yes
<i>international journal</i>	<i>professional journal</i>	no
<i>laws of florida</i>	<i>applicable laws</i>	no
<i>farnsfield</i>	<i>main settlements</i>	no
<i>ellroy</i>	<i>favorite authors</i>	no

Table 6: Interesting or questionable pairs.

# Evaluation: Expanding a Class

---

- Idea: expand the size of a given class through relaxation of constraints, i.e., less restrictive values for J/K for pre-specified label L
- Evaluating effects on quality: randomly select three classes based on size from three separate ranges
  - small: < 50 instances; selected classes: *prestigious private schools, telfair homebuilders, plant tissues*
  - medium: < 500; *goddesses, organisms, enzymes*
  - large:  $\geq 5000$ ; *flavors, critics, dishes*
- Required: each class should have shown increase in size of at least 50%
- Then, 50 instances were chosen randomly from minimum sized versions of each class, and maximum sized version of each class (removing those elements that are also contained in the min. sized classes)
- Results:
  - for small classes: 100% accuracy both before and after expansion; proves, that even small classes do not always necessary land together in distributionally similar clusters
  - if a class expands, then it happens that a class is incorrectly „spread out“ over clusters, leading to „nearly correct“ instances (increase of 40% to 66% for the class enzymes, 29% to 44% for the class goddesses)

# Evaluation: Handling Pre-nominal Adjectives

- Many of the classes (especially for small classes) had labels containing pre-nominal adjective modification.
- Table 8: ratio of labels with > 2 words, and 1st word an adjective ( $2^2$  means: S has between 4 and 8 instances; 455/936 means: 455 of the 936 classes have modified label names)
- Now: when removing modifier, classes can be merged, and size increase, but number of classes decreases, e.g., from 8572 to 3397, if adjective is in WN3.0
- Effect: also increase in quality of small classes (which means: few observed distributional similarity), because fewer incorrect labeling → GN: ako semantic underspecification

$S$	Ratio	%	$S$	Ratio	%
$2^{12}$	0/4	0%	$2^6$	342/961	36%
$2^{11}$	1/32	3%	$2^5$	627/1566	40%
$2^{10}$	4/73	5%	$2^4$	860/1994	43%
$2^9$	19/143	13%	$2^3$	852/1820	47%
$2^8$	68/259	26%	$2^2$	455/936	49%
$2^7$	170/541	31%	$2^1$	108/243	44%

Table 8: For each range, the number of classes with a label whose first term has an adjective reading.  $S = \lfloor size \rfloor$ .

Instance	Class
<i>lamborghini murcielago</i>	<i>real cars</i>
<i>spanish primera division</i>	<i>domestic leagues</i>
<i>dufferin mall</i>	<i>nearby landmarks</i>
<i>colegio de san juan de letran</i>	<i>notable institutions</i>
<i>fitness exercise</i>	<i>similar health topics</i>

Table 9: Examples of (instance, class) pairs sampled from amongst classes with less than 10 members, and where the label was deemed unacceptable.

# Evaluation: Task-Based

---

- Idea: Use the extracted classes as input for the task of extracting attributes
- Starting point (Paşca, 2007): Acquiring ranked list of class attributes from query logs based on a list of instances and seed attributes.
- Extension: Use J/K settings for determining the instances and ranking candidate attributes.
- Example result:
  - Label: *forages*, Instances: *alsike clover, rye grass, tall fescue, source lespedeza*; Attributes: *types, picture, weed control, planting, uses, information, herbicide, germination, care, fertilizer*
- Attribute precision (random sample of 25 classes): 70% at rank 10, 67% at rank 20

## Details in:

**Paşca, M., and Van Durme, B. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from Web documents and query logs. ACL-08.**

$$P = \left( \sum_{i=1}^N val(attr_i) \right) / N$$

$val(attr_i) = 1$ , if attribute  $attr_i$  is vital, 0.5 if ok, and 0 if incorrect