Entity Set Expansion

Günter Neumann, LT-lab, DFKI, Dec. 2011

NE set expansion and refinement

- Given a set of seed elements, automatically expand the set on basis of co-occurrence statistics.

- Given an automatically expanded NE set, refine this set by iteratively remove erroneous NEs from the set through feedback.

- A number of current approaches focus on extending list-based elements, and hence try to learn patterns and wrappers for the identification of enumerations in free text or in html.

Differences of approaches

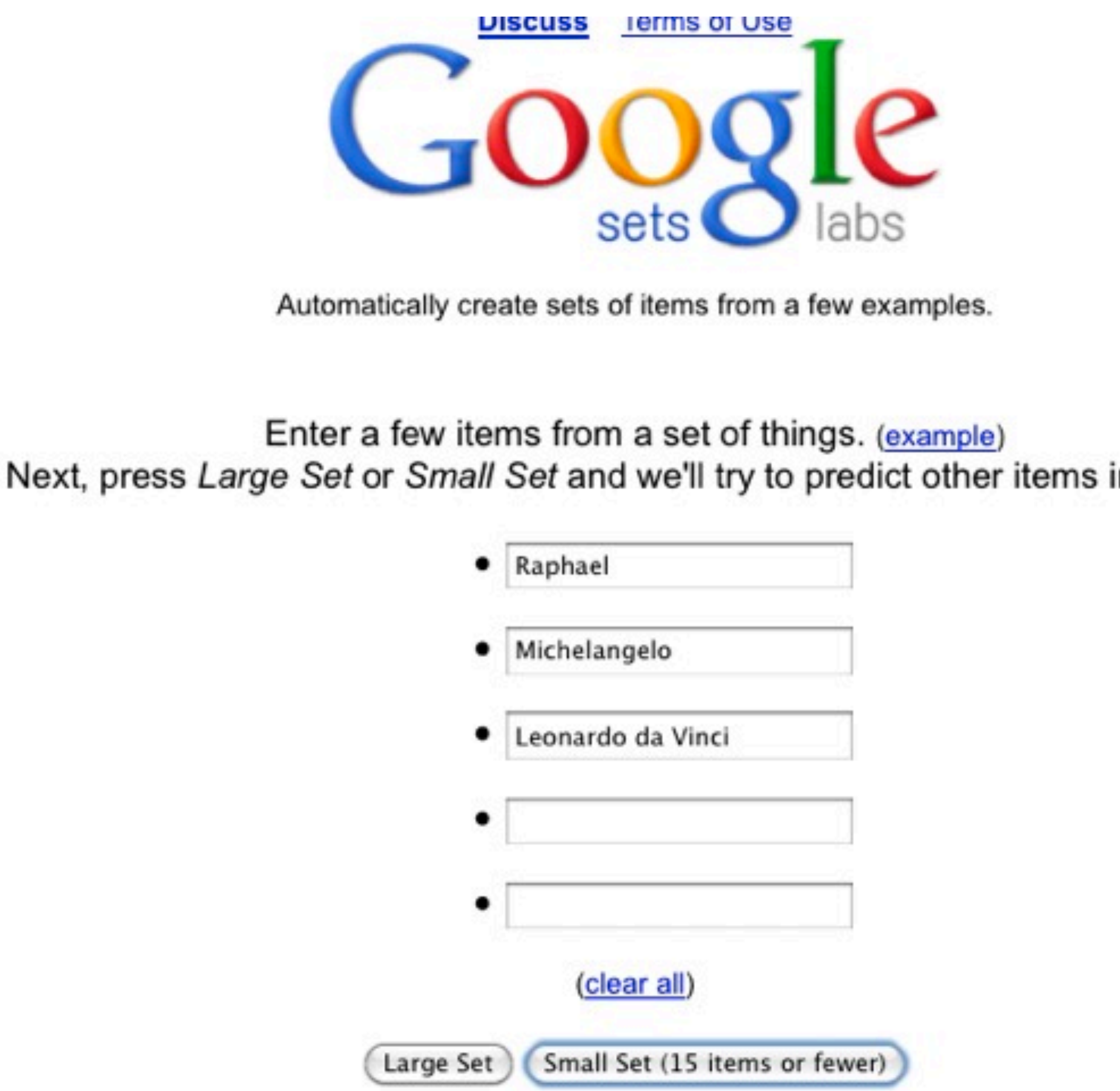
- Talukdar et al. 2006: grammar induction from text
- Sarmento et al. 2007: enumerations from texts
- Vyas & Patel 2009: as above, but learn to identify errors in expanded seed lists
- Van Durme & Pasca, AAAI-2008: Finding Cars, Goddesses and Enzymes: Parametrizable Acquisition of Labeled Instances for Open-Domain Information Extraction
- Wang&Cohen, 2007/2008: lists by automatically learning web-page specific wrappers; which are character level strings and hence completely language independent

NE set expansion

- General idea
  - given a small list of instances of some (**unknown**) class of entities
  - Consult a corpus in order identify similar entities to add to the class
- Example
  - seed set:
    - {Raphael, Michelangelo, Leonardo da Vinci}
  - expand set:
    - {Raphael, Michelangelo, Leonardo da Vinci, **El Creco, Sandro Botticelo, Jan van Eyck**}

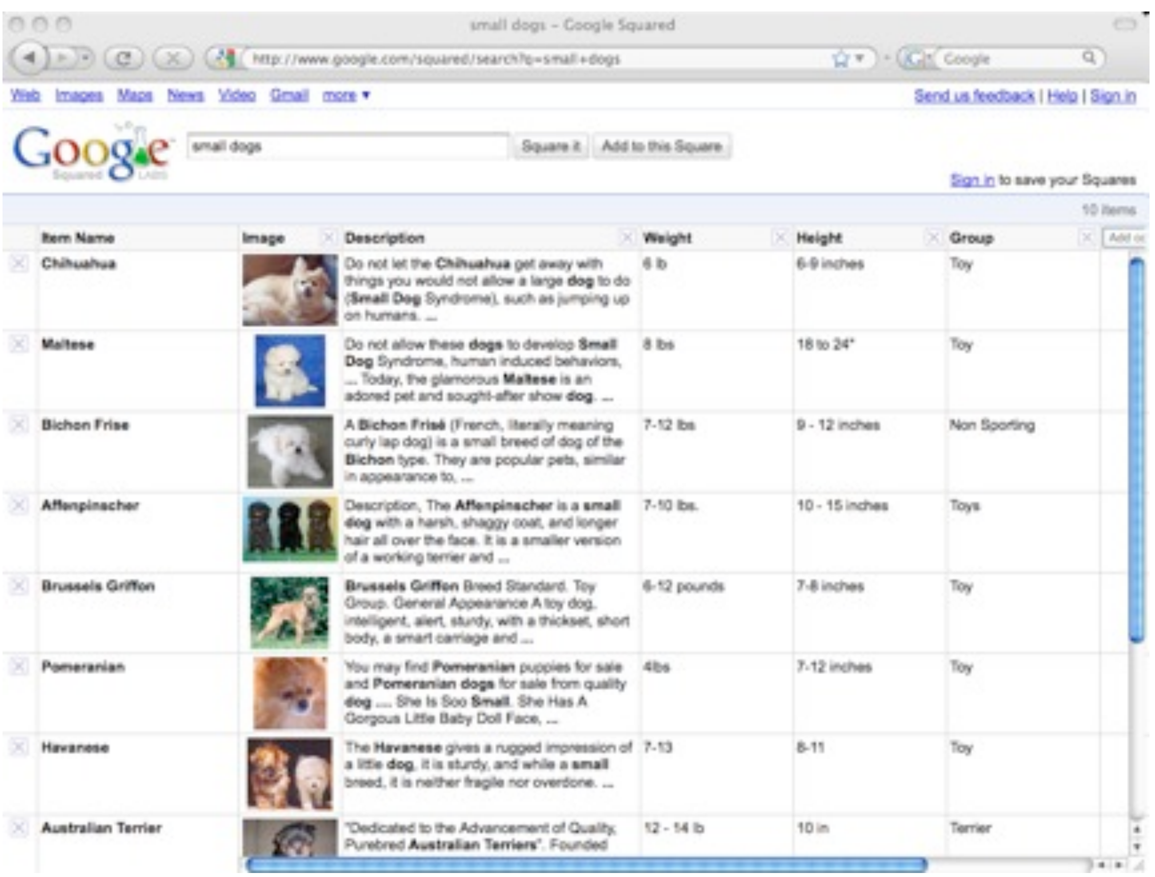
Example: Google Sets

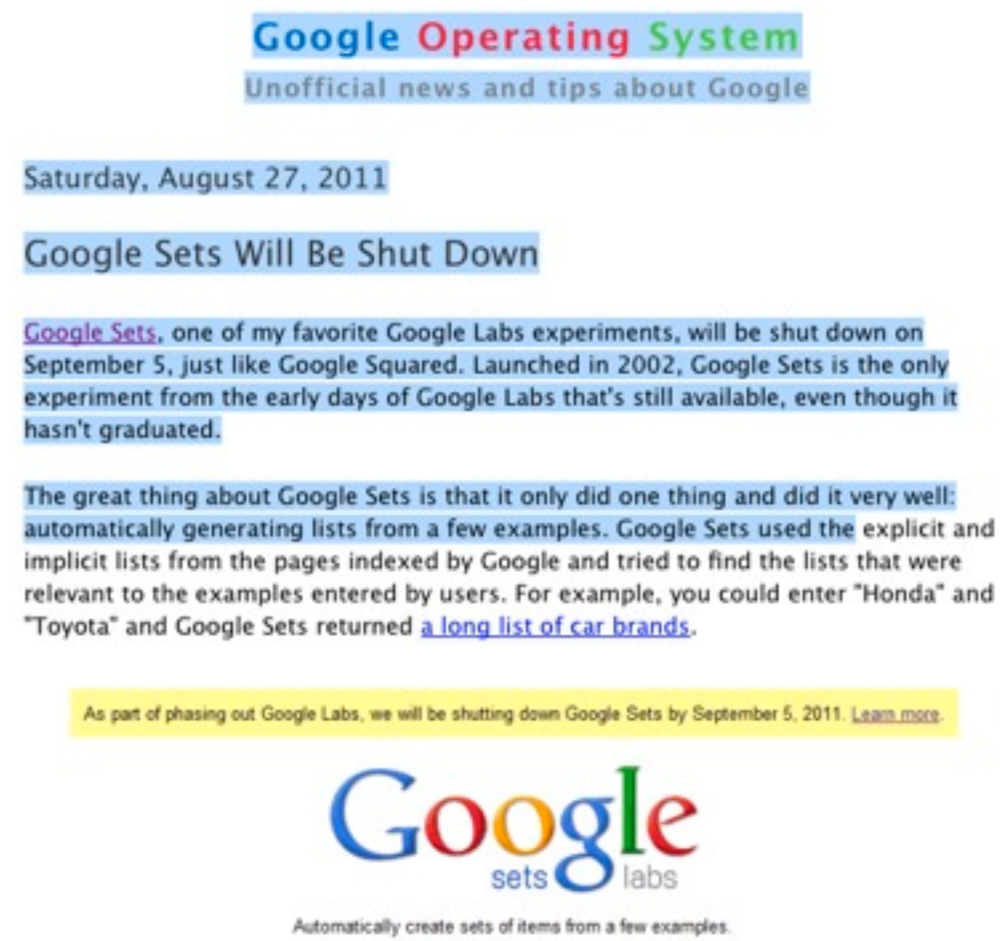
http://labs.google.com/set



Two further Issues Regarding Google Sets

- It has been shut down on 5th September 2011
- Google Sets was Used to Create Google Square





Google Operating System
Unofficial news and tips about Google

Saturday, August 27, 2011

Google Sets Will Be Shut Down

Google Sets, one of my favorite Google Labs experiments, will be shut down on September 5, just like Google Squared. Launched in 2002, Google Sets is the only experiment from the early days of Google Labs that's still available, even though it hasn't graduated.

The great thing about Google Sets is that it only did one thing and did it very well: automatically generating lists from a few examples. Google Sets used the explicit and implicit lists from the pages indexed by Google and tried to find the lists that were relevant to the examples entered by users. For example, you could enter "Honda" and "Toyota" and Google Sets returned a long list of car brands.

As part of phasing out Google Labs, we will be shutting down Google Sets by September 5, 2011. Learn more.

Google sets labs

Automatically create sets of items from a few examples.

- The good thing: Still used in Google Search to better understand the content of a page and to provide lists of related searches.

Sarmento et al. 2007: enumerations from texts

- A corpus-based approach to seed expansion
- Given a seed set, use **co-occurrence statistics** from a text corpus to define a membership function f()
- Rank NE candidates according to f()
- Evaluation framework that uses data from Wikipedia

NE set expansion

- Let:
  - S = set of seed entities of a class C
  - E = candidate entities
  - f(C,e) = x, where $0 \leq x \leq 1$ is the **degree of C-membership** of $e \in E$
- **Goal: learn f()**
  - approximate f(C,e) by f(S,e), and then compute f(S,e) for all $e \in E$
  - consider f(S,e) as similarity function between all seed elements and a single e.
  - compute f(S,e) by using a **vector space model**, i.e., by computing the similarity of feature vectors
- If we have several lists we can use them for POS and NEG

Vector Space Model for Seed Expansion

- Each element (i.e., each $s \in S$ or $e \in E$) w_j is represented by a vector of numerical features $vec(w_j)$.
- Given such a representation vec(), compare elements by standard distance functions, e.g. cosine.
- Choice of features defines the information that is captured and transferred to vec().
- For seed expansion: mainly type similarity

How to determine type similarity ?

- Observation: humans easily group and list type similar entities
  - „American Airlines' general rule is you can only bring one personal item such as a briefcase, purse or laptop bag on-board and one small piece of luggage."
  - The early British painters like Tilly Kettle, John Zoffany, John Smart, George Chinnary, William Hodges and others painted in oil.
- Goal: Identify such enumerations in text, and gather information about class similarity
- Assumption: if two elements consistently co-occur in lists, they are likely to be of a similar semantic class.

GN: similar to handle n-ary relation via combining binary relations

Identification of lists in texts

- Simple approximation: identify pairs of elements that belong to lists
- Assumption: lists are composed by sequences of pairs of coordinated words by
  - explicit coordinational elements (and, or, ...)
  - commas
- Look for text fragments likes:
  - „... ne_a, ne_b and ne_c ...“, „... ne_a, ne_b or ne_c ...“
  - „I lived in Paris, Berlin and London.“, „Experience with Java, C++ or Lisp is required.“
- Conclude: when instances of such patterns are found in text, then pairs (ne_a, ne_b) and (ne_b, ne_c) co-occur in coordination (for simplicity, no assumption for pair (ne_a, ne_c)).

Vector representation

- Main idea: represent candidates and seed elements as vectors encoding their co-occurrence frequency.
- Let NE ::= $\{ne_1, .., ne_N\}$ be all namend entities that co-occur in a text.
    - Define j-th component of $vec(ne_i) = |(ne_i,ne_j)|$
    - Similar: for seed set S, S(j) is defined as $|(s,ne_j)|$, $s \in S$
    - This means: a vector of a ne_i collects all ne_j, which co-occur with ne_i in an enmueration
- Two vector spaces
    - VS^x considers only pairs from explicit coordinations (more restricted, less noise, lower recall)
    - $VS^{‘}$ considers pairs from explicit and comma coordinations (more noise, e.g., „... X, Y ...“)

Membership function f()

ne_j \ ne_i	ne_1	ne_2	...	ne_N	s_1	...	s_m
ne_1	0	12		4	5		12
ne_2	3	0		11	3		6
ne_3	1	0		10	0		3
...							
ne_N	0	3		0	1		0

Consider the similarity of ne_j with **all** seed elements s_i as similarity between corresponding vectors.

use f^x or f' depending on vector space

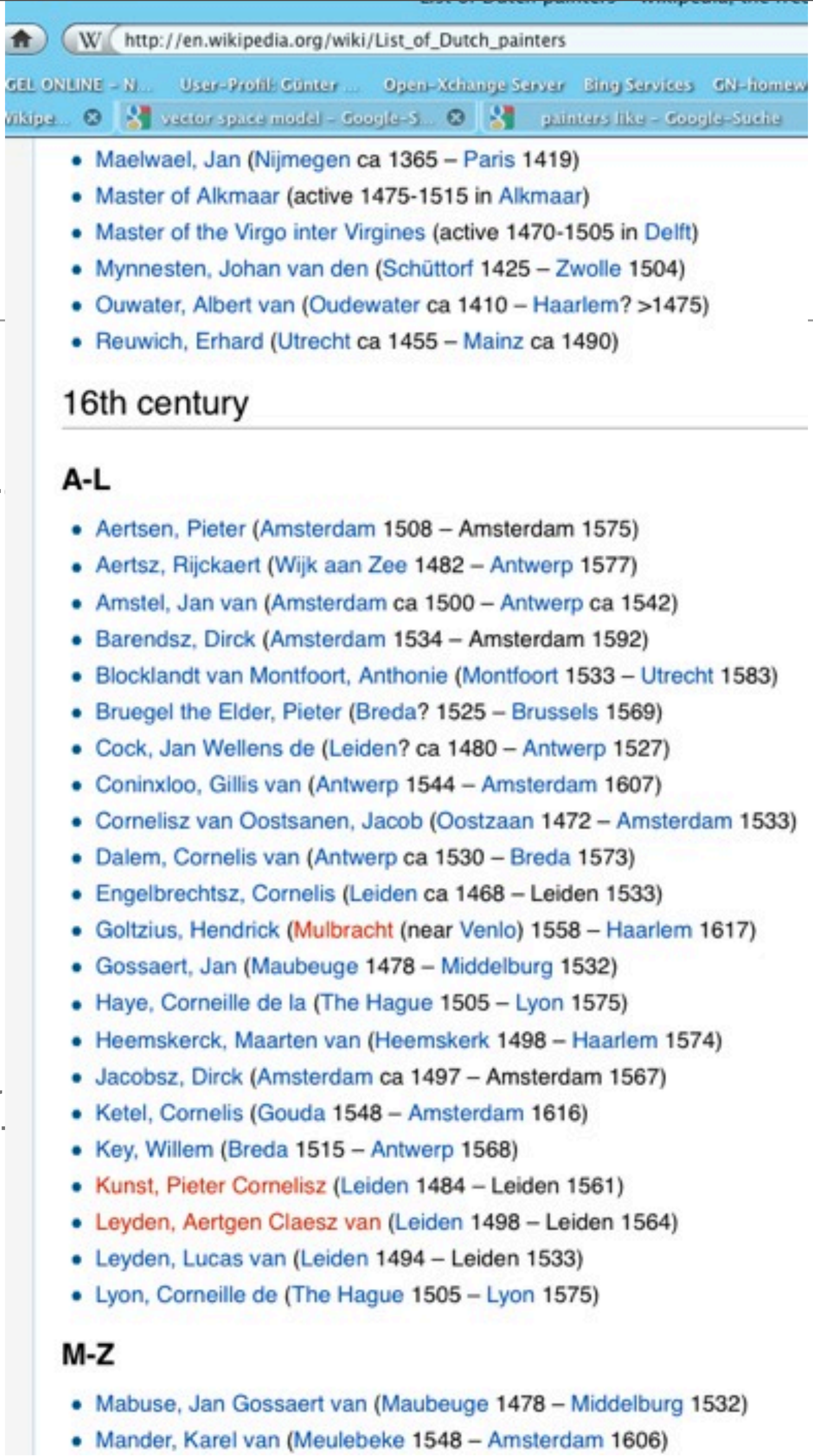
$$f(S,e) = \cos(vec(S),vec(e)) = \sum_{i=1...m} \cos(vec(s_i), vec(e))$$

$$\cos(vec(x),vec(y)) = (vec(x) * vec(y)) / (norm(x) * norm(y))$$

Evaluation using Wikipedia

- Wikipedia contains several explicit human-generated lists
  - gold standard for set expansion.
- A Wikipedia article is about a concept
- Simplified NE-identification in Wikipedia
  - when article A_1 contains a link L to an article A_2 and text(L) starts with capital letter, then text(L) is a mention of an entity, the one addressed by A_2
  - this approach bypasses problem of NE recognition

Evaluation using Wikipedia


http://en.wikipedia.org/wiki/List_of_Dutch_painters

- Maelwael, Jan (Nijmegen ca 1365 – Paris 1419)
- Master of Alkmaar (active 1475-1515 in Alkmaar)
- Master of the Virgo inter Virgines (active 1470-1505 in Delft)
- Mynnesten, Johan van den (Schüttorf 1425 – Zwolle 1504)
- Ouwater, Albert van (Oudewater ca 1410 – Haarlem? >1475)
- Reuwich, Erhard (Utrecht ca 1455 – Mainz ca 1490)

16th century

A-L

- Aertsen, Pieter (Amsterdam 1508 – Amsterdam 1575)
- Aertsz, Rijckaert (Wijk aan Zee 1482 – Antwerp 1577)
- Amstel, Jan van (Amsterdam ca 1500 – Antwerp ca 1542)
- Barendsz, Dirck (Amsterdam 1534 – Amsterdam 1592)
- Blocklandt van Montfoort, Anthonie (Montfoort 1533 – Utrecht 1583)
- Bruegel the Elder, Pieter (Breda? 1525 – Brussels 1569)
- Cock, Jan Wellens de (Leiden? ca 1480 – Antwerp 1527)
- Coninxloo, Gillis van (Antwerp 1544 – Amsterdam 1607)
- Cornelisz van Oostsanen, Jacob (Oostzaan 1472 – Amsterdam 1533)
- Dalem, Cornelis van (Antwerp ca 1530 – Breda 1573)
- Engelbrechtsz, Cornelis (Leiden ca 1468 – Leiden 1533)
- Goltzius, Hendrick (Mulbracht (near Venlo) 1558 – Haarlem 1617)
- Gossaert, Jan (Maubeuge 1478 – Middelburg 1532)
- Haye, Corneille de la (The Hague 1505 – Lyon 1575)
- Heemskerck, Maarten van (Heemskerk 1498 – Haarlem 1574)
- Jacobsz, Dirck (Amsterdam ca 1497 – Amsterdam 1567)
- Ketel, Cornelis (Gouda 1548 – Amsterdam 1616)
- Key, Willem (Breda 1515 – Antwerp 1568)
- Kunst, Pieter Cornelisz (Leiden 1484 – Leiden 1561)
- Leyden, Aertgen Claesz van (Leiden 1498 – Leiden 1564)
- Leyden, Lucas van (Leiden 1494 – Leiden 1533)
- Lyon, Corneille de (The Hague 1505 – Lyon 1575)

M-Z

- Mabuse, Jan Gossaert van (Maubeuge 1478 – Middelburg 1532)
- Mander, Karel van (Meulebeke 1548 – Amsterdam 1606)

- Wikipedia contains several explicit human-
  - gold standard for set expansion.
- A Wikipedia article is about a concept
- Simplified NE-identification in Wikipedia
  - when article A_1 contains a link L to an a capital letter, then text(L) is a mention o by A_2
  - this approach bypasses problem of NE

Evaluation using Wikipedia

- Wikipedia contains several explicit human-
  - gold standard for set expansion.
- A Wikipedia artic
- Simplified NE-ide
  - when article A capital letter, by A_2
  - this approach

http://en.wikipedia.org/wiki/List_of_Dutch_painters

- Maelwael, Jan (Nijmegen ca 1365 – Paris 1419)
- Master of Alkmaar (active 1475-1515 in Alkmaar)
- Master of the Virgo inter Virgines (active 1470-1505 in Delft)
- Mynnesten, Johan van den (Schüttorf 1425 – Zwolle 1504)
- Ouwater, Albert van (Oudewater ca 1410 – Haarlem? >1475)
- Reuwich, Erhard (Utrecht ca 1455 – Mainz ca 1490)

16th century

A-L

- Aertsen, Pieter (Amsterdam 1508 – Amsterdam 1575)
- Aertsz, Rijckaert (Wijk aan Zee 1482 – Antwerp 1577)
- Amstel, Jan van (Amsterdam ca 1500 – Antwerp ca 1542)
- Barendsz, Dirck (Amsterdam 1534 – Amsterdam 1592)

A **capital city** (or just **capital**) is the area of a *country, province, region, or state* regarded as enjoying primary status; although there are exceptions, a capital is almost always a city which physically encompasses the offices and meeting places of the seat of government and is usually fixed by law or by the constitution. An alternative term is **political capital**, but this phrase has a second meaning based on an alternate sense of the word *capital*. The capital is frequently the largest city of its constituent area.

The word *capital* is derived from the Latin *caput* meaning "head" and, in the United States, the related term *capitol* refers to the building where government business is chiefly conducted.

The seats of government in major sub-state jurisdictions are often called "capitals", but this is typically the case only in countries with some degree of Federalism, wherein major substate legal jurisdictions have elements of sovereignty. In unitary states, an "administrative center" or other similar term is typically used for such locations besides the national capital city. For example, the seat of government in a State of the United States is usually called its "capital", but the main city in a region of the United Kingdom is usually not called such, even though in Ireland, a county's main town is usually called its "capital". On the other hand, these four subdivisions of the United Kingdom do have capital cities: Scotland – Edinburgh, Wales – Cardiff, Northern Ireland – Belfast, and England – London. Counties in England, Wales and Scotland have historic county towns which are often not the largest settlement within the county and invariably no longer excercise and political power as the county is often only ceremonial and adminitrative boundries differ.

In Canada, the ten Provinces of Canada all have capital cities, including Quebec City, Toronto, Victoria, B.C., Winnipeg, et cetera. The states of such countries as Mexico, Brazil, and Australia all have capital cities. For example, the six state capitals of Australia are Adelaide, South Australia, Brisbane, Queensland, Hobart, Tasmania, Melbourne, Victoria, Perth, Western Australia, and Sydney, New South Wales. In Australia, the term "capital cities" is regularly referred to and includes the aforementioned state capitals plus the federal capital Canberra and Darwin, the capital of the Northern Territory.

In the Federal Republic of Germany, each of its constituent republics (or "Lands") has its own capital city, such as Wiesbaden, Mainz, Hamburg, Düsseldorf, Stuttgart, and Munich. Likewise, each of the republics of the Russian Federation has its own capital city.

At the lower administrative subdivisions in various English-speaking countries, terms such as county town, county seat, and borough seat are usually used.

Evaluation measure

- Starters
  - Let C = a Wikipedia list serving as gold standard
  - P ⊂ C of positive examples
  - N = negative examples, i.e., entities that are somehow related to entities in P, but that are not in C. Usually, N >> P
- Test case
  - Select seed set S ⊂ P
  - Construct candidates E = P ∪ N \ S
- Rank elements E by f(S,.), and asses quality of resulting ranking R using average precision

$$AP(S,R) = \frac{\sum_{r=1}^{|E|} P_{at}(r,R) \cdot \mathcal{I}(R(r) \in \mathcal{P} \setminus S)}{|\mathcal{P} \setminus S|}.$$

$$MAP = \frac{\sum_{i=1}^{m} AP(S_i, R_i)}{m}$$

$P_{at(r,R)}$ is value of P at rank r (i.e., number of positive examples in the ranked list R). Also called R-Precision, cf. next approach by Vyas&Pantel
I() is the indicator function of f()

Mean average precision for m test cases S_i

Construction of lists

- Wikipedia XML dump for English & Portuguese.
- Retrieve list with query „List of" and explicit HTML list structure (<ul>, <ol>; no tables).
- For each list element (ignore entirely numerical)
    - its frequency in Wikipedia (number of times element occurs as link)
    - its target Wikipedia article
- English: 17594 lists; av. size 92.4; av. size of linked elements 58.4; av. freq. 286.2
- Portuguese: 1390 lists; 90.3; 43.4; 32.5

Construction of P and N

- Only consider lists with at least m linked elements →ensure coverage
- For each list L(i) chose all items that are linked and that pass a frequency threshold → $P_{cand}(i)$
- For each list element extract all entities E(i) from their articles.
- Add each ne ∈ E(i) to $N_{cand}(i)$ if ne ∉ L(i) → consider only related entities as negative examples
- Define P(i) and N(i) as the topmost elements
- English: 3219 test sets; Portuguese: 75 (due to smaller Wikipedia)

Collection of pairs of coordinated entities

- From XML dumps, apply simple heuristics based on explicit coordination and commas and NE identification.
- Scan only texts in paragraphs with simple patterns
  - „(ne_a) (coordination connector) (ne_b)“

	NE Pairs	Distinct NE Pairs	Dim($\mathcal{VS}$)
$\mathcal{VS}'_{EN}$	2,172,790	1,255,204	819,379
$\mathcal{VS}^{\chi}_{EN}$	1,755,603	516,415	500,980
$\mathcal{VS}'_{PT}$	154,836	119,174	85,494
$\mathcal{VS}^{\chi}_{PT}$	44,919	36,751	46,601

Table 1: NE's extracted and Vector Spaces

Result

Same threshold was used, and hence actually relatively higher for Portuguese because Wikipedia is smaller; hence resulting lists contain more frequent elements.
Hence these tests with the least frequent elements chosen.

	#tests	f_{avg}	μ'	μ^{χ}
EN(all)	3219	1758.3	0.424	0.289
PT(all)	75	623.6	0.542	0.426
PT($\mathcal{P}_{28}$, $\mathcal{N}_{28}$)	28	982.2	0.547	0.493
PT($\mathcal{P}^{-}_{28}$, $\mathcal{N}_{28}$)	28	189.4	0.431	0.229

Table 2: Average values of MAP over all test sets

Highest result indicates that the performance of the membership function improves as the frequency of the elements to which they are applied increases.