ontologies

André Santos, afs@inesctec.pt

Disclaimer

Almost everything in the following slides has been shamelessly copied word for word from Alberto Simões' slides.

Classification

- From texts we get words
- Joining some words we get Named Entities
- Named Entities can be classified:
  - Person / Organization / Make or Product
- How to classify and characterize entities/individuals?

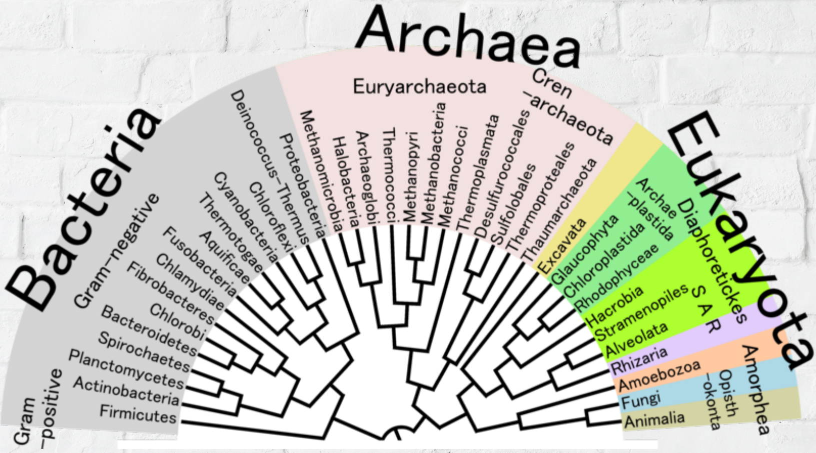
Classifying Individuals: Taxonomy

- Taxonomy is the practice and science of classification.
- Mathematically, a hierarchical taxonomy is a tree structure of classifications for a given set of objects.

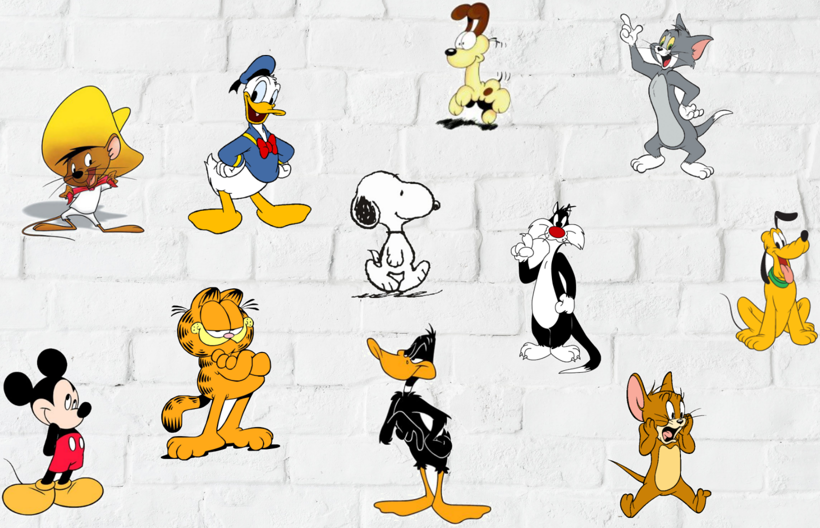
Classifying Individuals: Taxonomy

- Define two main types of relations:
  - Between Classes: class X is contained by class Y (and therefore, it inherits class Y's properties)
  - Between Individuals and Classes: e is one of X (e shares certain properties with other members of X)

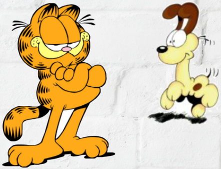
Taxonomy: the common example

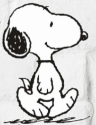


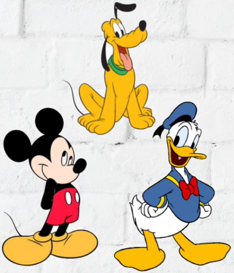
Taxonomy: Exercise



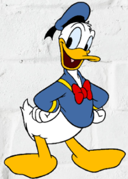
Taxonomy: Exercise







Taxonomy: Exercise



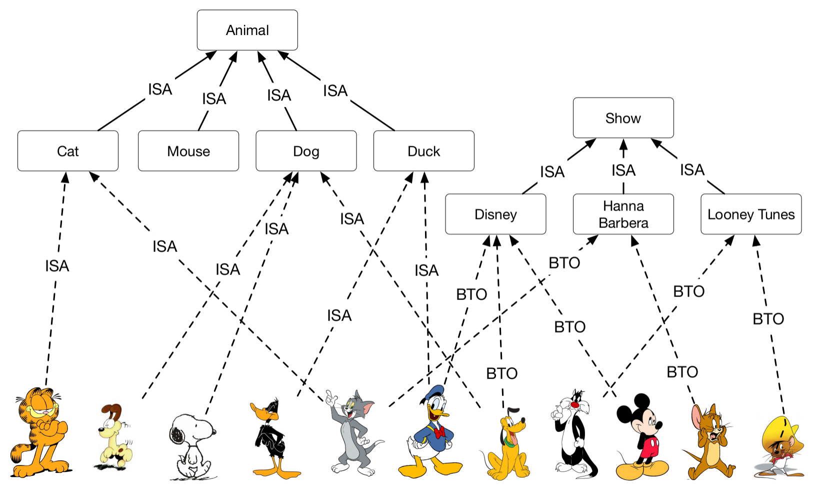
Taxonomy: Limitations

- Being an acyclic tree:
  - $\nexists C_i : C_i \subset C_a \wedge C_i \subset C_b$
    Classes can't inherit properties from two different classes
  - $\nexists e : e \in C_a \wedge e \in C_b$
    It is expected that individuals not be classified in two distinct classes

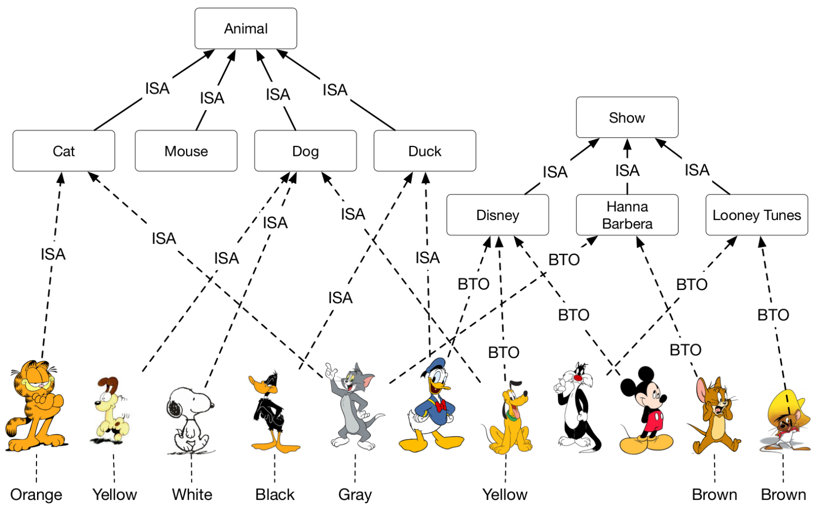
Taxonomy: Limitations

- Frequently, we need to represent overlapping classes:
    - Dolphins are mammals but live in the water
    - Restaurants may serve multiple types of food
    - Clothes may be unisex

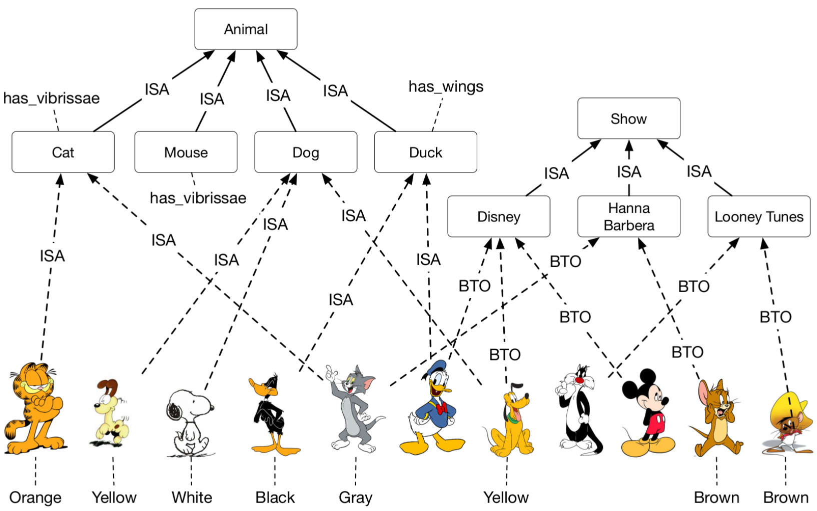
Ontology: multiple hierarchy



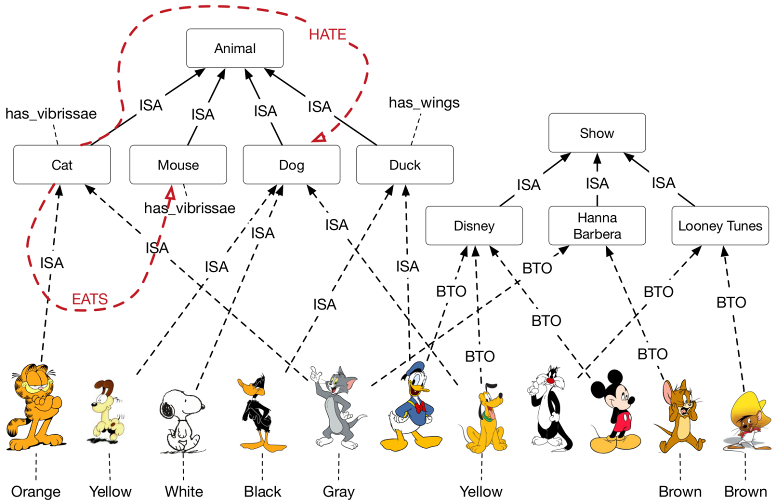
Ontology: instance properties



Ontology: class properties



Ontology: multiple relation type



Ontologies

What can we do with them?

- Several formats to represent them (OWL, SKOS, RDF [Turtle & N-Triples], ...)
- GUI: Protegé & webprotege.stanford.edu
    - reasoner
- Store: Apache Jena, 4Store, OpenLink Virtuoso, ...
- Query: SPARQL (SPARQL Protocol and RDF Query Language)

Practical assignment #3

- Due date: 19 Jan 2019
- By default, same groups
- Report + code + demo

Pick **one** of the options described on the next slides.

Option 1

Given a large annotated corpus, create a tool capable of calculating lemmas and POS tags for an isolated word or for word(s) in a sentence.

Option 2

Create a tool capable of, given a text where all spaces have been removed, re-add spaces to the text.

Bonus points: add more features, like making it capable of removing wrong spaces randomly added to the middle of words.

Option 3

Create a tool capable of correcting "tracinho se" errors in written Portuguese such as "estives-te" or janta-se / jantasse.

Option 4

Create a tool, OCRshot, to handle the following workflow:

1. take a screenshot of (some text on) your computer screen
2. add some meta information
3. run an OCR tool on that screenshot
4. post-process the resulting text according to the meta information added
5. produce some output objects

Option 5

Create a tool, inoti-make, which is an `inotify`-based version of a makefile:

define patterns and folders to be watched by `inotify` and functions/scripts to be executed in reaction to those events.

Option 6

Create a spell checker for Mbundu (Umbundu/Kimbundu), a group of languages spoken in Angola.

Study the morfological rules of these languages, normalize a corpus and produce a list of words to be used to feed aspell, hunspell and/or jspell.

Option 7

Fetch text documents from a website (hint: pick one with an RSS feed), pre-process them, index them and implement a search functionality using the TF-IDF algorithm.

Option 8

...

Proceed with caution

- Tell us which option your group will be doing (email)
- **Come and talk to us before starting!**
- Assignment descriptions are vague
- Most of the options need a brainstorm before begining
- We can help narrowing the scope of the assignment to make it feasible
    - ...or the inverse :)

And also

- Bonus points for
  - dealing with large ammounts of text
  - calculating performance metrics (precision, recall, ...)