Escarafunchando* o sistema KnowItAll

Simpósio Doutoral - Linguateca

Marcirio Silveira Chaves

* fig., investigando, procurando com minúcia.

Estrutura

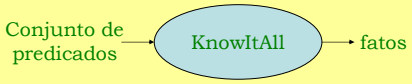
- Panorâmica
- Principais Componentes
  - Inicialização
  - 2 Principais Módulos
- Métodos utilizados para melhorar a abrangência
  - Aprendizado de padrões
  - Extração de subclasses
  - Extração de listas
- Resultados

Panorâmica

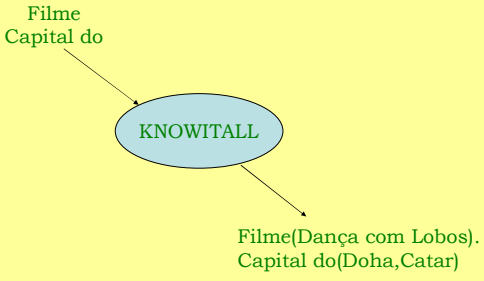
- Extracção de grandes coleções de fatos, termos e relacionamentos da web
  - Não supervisionado
  - Independente do domínio
  - Modo escalável
- Objetivo: melhorar a abrangência sem sacrificar a precisão

Panorâmica

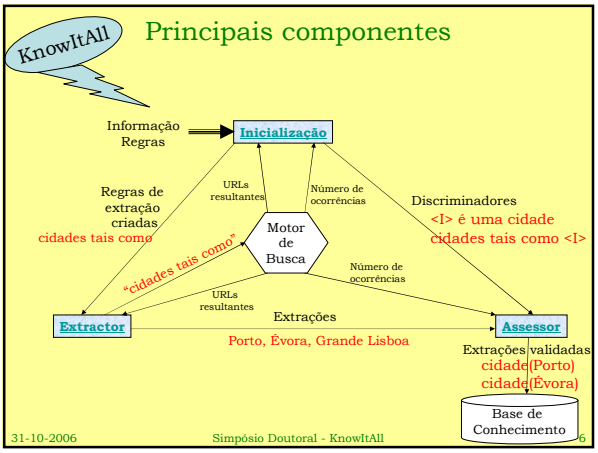
- Atribui uma probabilidade a cada fato extraído



Panorâmica



Principais componentes



Inicialização

- Uso de padrões

  Ex. de regra:
  Predicado: Class1
  Padrão: NP1 "tal como" NPList2
  Restrições: head(NP1)=plural(label(Class1))
  properNoun(head(each(NPList2)))

  Ex. de instância:
  Predicado: Cidade, município
  Padrão: NP1 "tal como" NPList2
  Restrições: head(NP1)="cidades", "municípios"
  properNoun(head(each(NPList2)))

Inicialização

- Uso de padrões
- Ex.:

  Class "tal como" **NPList**
  cidade tal como
  cidades tais como
  município tal como
  municípios tais como

2 Módulos Principais

- *Extractor*
  - Utiliza um analisador sintático
  - Cria uma consulta a partir das palavras-chave em cada regra
  - Envia a consulta para o motor de busca
  - Aplica a regra para extrair informação das páginas resultantes
  - Testa se o *head* de cada *NP* da lista é um nome próprio
  - Extrai o *head*

    • Extrai:
    cidades tais como **Porto**, **Braga** e **Guimarães**

    • Não extrai:
    Mapas detalhados e informação para várias cidades tais como **mapas de aeroportos**, **centro das cidades**, etc.

2 Módulos Principais

- *Assessor*
  - Discriminadores (validadores)
    - Submetem os extratos gerados pelo *Extractor* ao motor de busca
    - Ignora pontuação, espaço em branco e etiquetas HTML
    - Se resultados válidos, insere na base de conhecimento
  - Ex.: *slot string – imediatamente adjacentes*

Predicado Unário	Predicado Binário
Cidade Lisboa	Sintra Concelho de Lisboa
Concelho Porto	Chaves parte de Vila Real
Freguesia Santa Isabel	

2 Módulos Principais

- *Assessor*
  - Discriminadores (validadores)
    - PMI = Pointwise Mutual Information
    - D = Discriminator
    - I = Instância

$$PMI(I,D)=\frac{|Hits(D+I)|}{|Hits(I)|}$$

$$PMI(I,D)=\frac{|Hits(cidade+Évora)|}{|Hits(Évora)|} \qquad PMI(I,D)=\frac{94.700}{878.000}=0{,}11$$

Problema: homônimos

Modelo com aprendizados

Métodos utilizados para melhorar a abrangência

- Aprendizado de padrões
- Extração de subclasses
- Extração de listas

Aprendizado de padrões

- Recebe um conjunto de instâncias geradas a partir do *Extractor*
- Consulta a web
- "melhores padrões são selecionados" = **Padrões aprendidos**

Objectivo: Encontrar padrões de alta qualidade

Aprendizado de Padrões

- Classe <*cidade*> e conjunto de instâncias (Lisboa, Algarve, Vila Nova de Gaia)
- Procura as intâncias na web e guarda contexto
  - (w w w w <I> w w w w)
- Identifica padrões candidatos

  Ex.: l**ocalizado em** Lisboa
  está situado no Algarve **a 40 Km de**
  perto de Vila Nova de Gaia **no norte do país**

- Avalia os padrões (Precisão estimada)

Aprendizado de padrões

❑ Padrões aprendidos atuando como Extratores

- **localizado em** <cidade>
- **situado no** <cidade>
- **perto de** <cidade>

Consulta a web.

- Qualquer nome próprio ocorrendo após **os padrões aprendidos** torna-se um candidato a cidade.

Aprendizado de padrões

❑ Padrões aprendidos atuando como Discriminadores

- <I> é uma cidade
- Limiar aprendido: 0.000016

- cidades tais como <I>
- Limiar aprendido: 0.000044

- <I> e outros municípios
- Limiar aprendido: 0.000032

- cidades incluindo <I>
- Limiar aprendido: 0.00027

- cidades <I>
- Limiar aprendido: 0.0000078

Segundo método para aumentar a abrangência do KnowItAll

Extração de subclasses

Extração de subclasses

– Padrões Hearst

Ex.:

Padrão	Extração
C tal como N	éUm(N,C)

– Avaliando subclasses candidatas
  - Análise morfológica do prefixo
    – ***Microbiologist*** *é uma subclasse de* ***Biologist***
  - Verifica no WordNet
  - Se ocorre, assume uma alta probabilidade ao fato encontrado

Extração de subclasses

- Independente do contexto
  – Cientista, Cidade e Filme
  Ex.: "cientistas tais como"
- Dependente do contexto
  – Contexto = domínio farmacêutico
  – Pessoas, Produtos e Organizações
  – Consulta na web
    - Regra + palavra-chave relevante do domínio

  Ex.:"people such as" ***pharmaceutical***

Extração de subclasses

- Termos (exportador, fornecedor, etc.) que se referem para ambas as classes **Pessoa** e **Organização**, no **WordNet** aparecem somente como subclasses de **Pessoa**.

Terceiro método para aumentar a abrangência do KnowItAll

Extração de listas

Extração de listas

- Métodos anteriores
  – Texto não estruturado
- Extração de listas
  – Premissa:
    - Muitos sítios web são gerados automaticamente a partir de bases de dados
  – Estrutura regular de página web
  – Uso de etiquetas HTML

Extração de listas

- Entrada:
  – Nome de uma classe + "Sementes Positivas"

  Ex.: Rio + Tejo, Douro
- Saída
  – Conjunto de termos candidatos para a classe

  Ex.: Tejo, Douro, Mondego, Ria de Aveiro, Ave, Cávado

Extração de listas

- Observações sobre os resultados
  - Negativas
    - Aeroportos, hotéis e países são frequentemente listados com cidades
  - Positivas
    - Itens encontrados não disponível em forma de texto

Resultados

- Resultados

Classe	Método	Precisão
Cidade	Inicialização	83%
Cidade	Aprendizado de padrões	66%
Cidade	Extracção de subclasses	70%
Cidade	Extracção de listas	52%