Empoderando vidas.
Fortalecendo nações.

Projeto BRA/12/018 - Desenvolvimento de Metodologias de Articulação e Gestão de Políticas Públicas para Promoção da Democracia Participativa

Produto 03 - Ferramentas assistidas de categorização de conteúdo

Com base em Processamento de Linguagem Natural e de Redes Complexas, adaptadas para o ambiente do portal de participação

Renato Fabbri

Secretaria Geral da Presidência da República

Produto 03 - Ferramentas assistidas de categorização de conteúdo

Contrato n. 2013/000566

Objeto da contratação: Aporte de conhecimentos e tecnologias para especificação de vocabulário e ferramentas assistidas que utilizam processamento de linguagem natural e análise de redes complexas para o conteúdo do portal da participação social.

Valor do produto: R$ 10,800 (dez mil e oitocentos reais)

Data de entrega: 27 Julho de 2014

Nome do consultor: Renato Fabbri

Nome do supervisor: Gabriella Vieira Oliveira Gonçalves

Secretaria Geral da Presidência da República

Fabbri, Renato

Ferramentas assistidas de categorização de conteúdo: Com base em Processamento de Linguagem Natural e de Redes Complexas, adaptadas para o ambiente do portal de participação / 2014.

Total de folhas: 15

Supervisor: Gabriella Vieira Oliveira Gonçalves

SG/PR

Secretaria Geral da Presidência da República

Palavras-chave: reconhecimento de padrões, redes complexas, processamento de linguagem natural, participação social.

Sumário

Resumo

Este documento descreve procedimentos selecionados para categorização de conteúdo do portal federal da participação social (Participa.br). O produto relacionado no termo de referência desta consultoria preve somente propostas de especificações e códigos. Dado o aspecto prático do trabalho, estão descritas o contexto e possibilidade consideradas, asism como implementações e códigos operantes. Parte deste trabalho é acessível online via http, como os scripts no IPython Notebook e o endpoint SparQL que serve os dados do Participa.br com critérios semânticos.

Palavras-chave: reconhecimento de padrões, redes complexas, processamento de linguagem natural, participação social.

1 Introdução

1.1 Contexto e importância da consultoria

Em confluência com o portal federal de participação social (Participa.br), e o Plano Nacinal de Participação Social (PNPS), esta consultoria propõe métodos de classificação e priorização de conteúdo, e formas de auto-regulação para o portal. O presente produto apresenta uma seleção de métodos para classificação de conteúdo. Dadas a pertinência para o contexto participativo e a simplicidade, são apresentadas a classificação via 1) conectividade dos autores e 2) características dos documentos.

1.2 Contexto e importância do Produto

- Este Produto, através da classificação de conteúdos, visa 1) facilitar a assimilação das informações pelos participantes; 2) explicitar propriedades do sistema considerado; 3) permitir observação de conteúdos produzidos por nichos ou características em comum.
- São esperadas a incorporação destes métodos no funcionamento do Participa.br e pelos participantes.
- A especialização conectiva dos agentes sociais, e do texto produzido por indivíduos e grupos, é um fenômeno plenamente reconhecido. O aproveitamento destas diferenciações é uma realidade, mesmo ainda restrito a empresas e acadêmicos. A entrega prática destes conhecimentos ao poder federal e à sociedade capacita a democracia participativa.

2 Desenvolvimento

2.1 Etapas de desenvolvimento

2.1.1 Estudo ontológico e triplificação dos dados para API de acesso

Para viabilizar a classificação de conteúdos do portal participativo, em confluência com as propostas de web semântica desta consultoria e do Participa.br, foi necessário uma abordagem ontológica dos aspectos envolvidos no participa.br, assim como a triplificação dos dados. Para isso, a OPS (Ontologia de Participação Social) foi revisada, com melhoras substanciais, também a OPA (Ontologia do Participa.br) foi criada[1,2]. Já a representação dos dados do Participa.br em triplas RDF envolveu o uso destas e diversas outras ontologias[3].

2.1.2 Instanciação de um Fuseki/Jena e um IPython Notebook

Os dados triplificados podem ser usados diretamente em algum aplicativo ou programa. A forma padrão de disponibilizar dados em RDF, porém, é através de um endpoint, que prepara os dados na RAM para buscas especificadas via SparQL. Está online um endpoint Jena para consultas SparQL via HTTP. Também uma seleção dos scripts em Python estão disponíveis através de navegadores comuns, como o Firefox ou o Chrom(e,um). **Veja os anexos**.

2.1.3 Classificação dos textos (mineração de texto / processamento de linguagem natural)

As possibilidades de classificação de conteúdo com base nos textos são inúmeras. Nesta subsubseção, são apontados alguns dos caminhos considerados.

- Através de uso de textos previamente classificados, pode-se treinar classificadores automatizados. Este é o chamado "aprendizado supervisionado" de máquina. As técnicas atualmente em uso são inúmeras (redes neurais, algorítmos genéticos, etc). Para exemplificação, foi implementado uma aprendizagem Bayesiana. **Veja a implementação nos anexos**.

- A classificação de objetos sem classes previamente definidas, com base somente nas propriedades dos objetos, é conhecido como "aprendizado não supervisionado". Pode-se impor a existencia de 2 classes (com base no balanço estrutural[easley]), ou mais classes, de forma a maximizar a dispersão inter-classe e diminuir a dispersão intra-classe. Este processo pode ser útil para observar nichos nas atividades, mesmo sem um conjunto de mensagens classificadas de antemão.

- Classificação de mensagens similares às escolhidas. Esta distância pode ser euclidiana no espaço de contagem de palavras, ou calculada via redes semânticas (e.g. wordnet).

- Classificação via contexto similar da palavra ou via simples incidência da palavra. Como buscadores usuais, com capacidades mais amplas para lidar com contexto (outras palavras, tipo de autor, classificação da mensagem).

- Ranqueamentos para mensagens, autores e palavras:

  - Mais adjetivos, mais substantivos, mais pontuações, etc.
  - Maior tamanho médio das palavras ou variedade de tamahos (desvio padrão). **Ver nos anexos**.
  - Frases mais longas em caracteres ou em palavras, variedade de tamanhos (desvio padrão).

– Uso de limiares para o rankeamento, p.ex.: os participantes que mais usam adjetivos (ou escrevem mensagens de mobilização) dentre os que possuem mais de 10 mensagens. **Ver nos anexos**.

–

2.1.4 Classificação dos agentes pela conectividade (Redes Complexas)

- Em geral, as redes formadas com rastros de atividade digital são: redes de interação ou redes de relações. No participa, há, em especial, a rede de amizades entre os usuários (relações) e redes de interação: quem responde quem, etc. **Ver nos anexos**.
- Pode-se classificar os usuários por comunidades detectadas nas redes. **Ver nos anexos**.
- Ranqueamento por centralidade é um dos recursos mais comuns. Há medidas de centralidade com base da conectividade (grau), intermediação (betweeness) proximidade (closeness) e outras medidas. **Ver nos anexos**.
- As redes sociais, por serem em geral "livres de escala", possuem especialização dos agentes, canonicamente pensado em "hubs", "intermediários" e "periféricos". Estes setores podem ser obtidos com mais propriedade comparando o histograma de conectividade da rede real com uma Erdös-Renyi com o mesmo número de vértices e arestas. **Ver nos anexos**.

2.1.5 Combinação de medidas de RC, PLN e outras

- As medidas de redes e de texto podem ser combinadas para melhorar a qualidade dos classificadores de mensagem. As estabilidades nestas medidas sugerem que hajam outliers[]]].
- Medidas de uso do sítio e do perfil do participante podem enriquecer os classificadores.

2.1.6 Aquisição de dados classificados

Para o aprendizado supervisionado (etiquetação automática, análise de semtimento, etc), é utilizado um conjunto de dados etiquetados de antemão, para "treinar" o classificador. Nas áreas de comunicação e monitoramento, são etiquetadas à mão as mensagens como positivas, negativas e neutras e em outras classes de interesse (e.g. geolocalizações, assuntos). Os autores são classificados em personas (autor masculino, feminino, ativista, militante, curioso, etc). Esta classificação manual pode servir para treinar um classificador, especialmente se revisada por uma ou mais pessoas.

2.2 Justificativa do método

- Classificações mais fundamentais: os métodos utilizados (bag-of-words, aprendizado bayesiano, medidas de grau e betweenness) são as mais usuais, além de facilitar a comparação e estabelecimento de benchmarks, possuem eficiência conhecida e significados mais facilmente compartilhados.

- Amadurecimento com equipe do Participa.br: há outros consultores e integrantes da SG/PR, e da sociedade civil, que compõem ou se comunicam com a equipe do Participa.br. Neste contexto, foram propostas e amadurecidas diversas possibilidades de classificação de conteúdos. Neste processo, foi decantado esta seleção, apresentada neste Produto.

2.3 Justificativa das fontes

- Pesquisa científica: o autor é pesquisador nas áreas relacionadas com produção bibliográfica em revistas internacionais e em circulação nacional.

- Os frameworks computacionais utilizados (nltk, networkx, rdflib, jena, etc) são amadurecidos no mundo todo, em desenvolvimento aberto, com comunidades em constante e pública discussão.

- A equipe do Participa.br é a equipe da SG/PR voltada para a participação social. Desta equipe provém boa parte dos avanços na participação social.

2.4 Confronto entre os resultados esperados e os alcançados

Este Produto preve "propostas de especificações e códigos" de classificação de conteúdo do Participa.br. Este Produto compreende estas propostas. Há, além disso, alguns resultados alcançados a mais:

- Propostas operantes em códigos online, já integrado aos dados semânticos e disponibilização via endpoint SparQL.

- Interfaces/frontends já estudadas para gráficos, reatividade e streaming (meteor+d3) [][][].

- Entrega, através dos resultados dos scripts, de uma breve análise do Participa.br em termos dos rastros digitais, dos conteúdos e dos usuários. **Ver anexos**.

Este documento e os scripts foram reunidos em um repositório git público usual[].

3 Usos dos resultados

O próximo Produto desta mesma consultoria possui foco na utilização destas classificações. Exemplos de usos estão topificados abaixo.

- Navegação dos conteúdos do portal: facilitar a aquisição das informações de interesse; permite observar o conteúdo com base em características dos participantes (e.g. hub, periférico, intermediarios) ou dos conteúdos (e.g. fração de adjetivos ou classificada com rótulos X ou Y).
- Enriquecimento do legado semântico do Participa.br e outros portais: boa parte dos cálculos, necessário para obtenção das estatísticas e classificações, requerem recursos computacionais poderosos e técnicas nada triviais. Assim, os resultados podem ser disponibilizados junto aos dados, em RDF.
- Atribuição de função: através das estatísticas dos grupos, pode-se recompensar atores ou convidá-los para atividades ou funções especiais.
- Resumos: usualmente dashboards, redes ou relações de palavras, visões gerais da entidade de interesse. A entidade pode ser um portal, uma comunidade, um usuário, uma trilha participativa. Estes resumos são bastante úteis para valorizar as instâncias e orientar os participantes.
- Coleta destas informações para usos/ações: difusão de mídia, consultas, propostas, estudos, etc.

4 Conclusão

A categorização de conteúdo do Participa.br pode ser feita de forma distribuída. Os dados, servidos por um endpoint SparQL, podem ser analisados por frontends, como um IPython Notebook, um ScrapperWiki ou um Meteor+d3 para visualizações interativas. Foi testado um intermediário em Flask para servir os dados já formatados para o frontend. Funcionou com serviços gratuitos do Heroku, Meteor e Mongo Labs, embora com limitações e alguns impasses para desenvolvimento em nuvem. Os anexos ao final deste documento[], e o repositório git[] são os resumos principais do Produto.

4.1 Comentários, sugestões, recomendações

- Para boas aplicações de classificadores de conteúdo, é necessário uma quantidade grande de conteúdo classificado previamente, geralmente à mão. Assim, é pertinente a etiqeutação das

mensagens com os parceiros da comunicação, para liberação junto aos dados semânticos e treino de classificadores.

4.2 Impacto do Produto para a elaboração, gestão e/ou avaliação de políticas públicas de participação social

- Facilita a apropriação dos processos participativos através da categorização de conteúdos e observação de suas características.
- Explicita a entrega das informações para a população, para observação distribuída.
- Entrega em tecnologias livres destes conhecimentos. Compostos por tecnologias livres e publicados em licença livre.
- Entrega de instâncias operantes de acesso aos dados do Participa.br, em formato RDF e enriquecidos.
- Semanticamente (OWL, OPS, OPA, FOAF, Dublin Core, etc).[]
- Entrega destes algoritmos em forma executável em navegadores HTTP comuns, como Firefox ou Chrome(um).[]

4.3 Como o Produto deverá impactar o público-alvo das políticas públicas a que se refere

- Permitindo navegação seletiva pelos conteúdos disponíveis.
- Valorizando as instâncias e as tornando mais informativas, com resumos estatísticos e visuais.
- Explicitando propriedades dos processos participativos e usos destas propriedades.
- Integrando o portal federal de participação social (Participa.br) ao legado humano de dados linkados (via critérios semânticos).
- Permitindo critérios funcionais para atribuição de papéis para participantes. Por exemplo, a construção de um manifesto ou resumo final pode ser feito requisitando: de periféricos, os substantivos; de hubs, os adjetivos; e de intermediários, que formem o texto com aquelas palavras. Outra possibilidade é a remuneração de hubs pela participação efetuada ou a convocação de periféricos para oxigenar o processo participativo.

- Aproximando perfis técnicos pela qualidade das tecnologias utilizadas e da relevância dos dados sobre participação social.

Referências

[1] "Single sign-on - wikipedia," http://en.wikipedia.org/wiki/Single_sign-on - Acessado em 22 de Maio de 2014.

Abreviações e jargão

OPS: Ontologia de participação Social

OPA: Ontologia do Participa.br

MMISSA: Monitoramento Massivo e Interativo da Sociedade pela Sociedade para Aproveitamento

AARS: A Análise de Redes Sociais

PNPS: Plano Nacional de Participação Social

RDF: Resource Description Framework

HTTP: Hypertext Transfer Protocol

SPARQL: Simple Protocol and RDF Query Language

endpoint SPARQL: ponto de acesso, geralmente HTTP, a dados em RDF via buscas em SPARQL.

Participa.br: Portal federal de participação social.

IPython Notebook: instância online para rodar scritps Python

Mateor: arcabouço para páginas reativas e com funcionamento distribuído.

D3js: biblioteca de visualização de dados.

Anexos

1. Exemplo em código computacional de classificação de conteúdo via texto
2. Seleção por rankeamento (tamanho de palavra) e limiar (número mínimo de palavras)
3. Criação de redes de amizade e de conteúdo do Participa.br
4. Deteção de comunidades
5. Ordenação (ranking) por centralidade
6. Setores da ordenação (ranking)
7. Exemplo em código computacional de classificação de conteúdo via conectividade dos participantes.
8. Script para testar o tempo de resposta do endpoint SparQL como conexão local e remota. OK
9. Experimentos online.

1. Exemplo de classificação de conteúdo via texto

Importação das bibliotecas para importação dos dados semânticos:

In [70]:
```
from SPARQLWrapper import SPARQLWrapper, JSON
import time
```

Definição de prefixos úteis para as triplas rdf:

In [71]:
```
PREFIX="""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ops: <http://purl.org/socialparticipation/ops#>
PREFIX opa: <http://purl.org/socialparticipation/opa#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX tsioc: <http://rdfs.org/sioc/types#>
PREFIX schema: <http://schema.org/>"""
```

Buscando todos os comentários no endpoint SparQL disparado pelo Jena:

In [72]:
```
NOW=time.time()
q="SELECT ?comentario ?titulo ?texto WHERE \
           {?comentario dc:type tsioc:Comment.\
          OPTIONAL {?comentario dc:title ?titulo . }\
           OPTIONAL {?comentario schema:text ?texto .}}"
sparql3 = SPARQLWrapper("http://localhost:82/participabr/query")
sparql3.setQuery(PREFIX+q)
sparql3.setReturnFormat(JSON)
results4 = sparql3.query().convert()
print("%.2f segundos para puxar todos os comentários do Participa.br"%
        (time.time()-NOW,))
```

```
2.43 segundos para puxar todos os comentários do Participa.br
```

Removendo pontuação e fazendo lista de palavras:

In [73]:
```
msgs_=results4["results"]["bindings"]
msgs=[mm for mm in msgs_ if ("titulo" not in mm.keys()) or
      (("teste de stress" not in mm["titulo"]["value"].lower())
      and ("comunidade de desenvolvedores e nesse caso, quanto mais"
           not in mm["texto"]["value"].lower()))]
exclude = set(string.punctuation+u'\u201c'+u'\u2018'+u'\u201d'+u'\u2022'+u'\u2013')
NOW=time.time()
import string, nltk as k
palavras=string.join([i["texto"]["value"].lower() for i in msgs])
palavras = ''.join(ch for ch in palavras if ch not in exclude)
palavras_=palavras.split()
print(u"feita lista de todas as palavras de todos os comentários em %.2f"%
      (time.time()-NOW,))
```

```
feita lista de todas as palavras de todos os comentários em 0.19
```

Removendo stopwords e fazendo contagem das palavras restantes:

In [74]:

```
NOW=time.time()
stopwords = set(k.corpus.stopwords.words('portuguese'))
palavras__=[pp for pp in palavras_ if pp not in stopwords]
fdist_=k.FreqDist(palavras__)
print("retiradas stopwords feita contagem das palavras em %.2fs"%
      (time.time()-NOW,))
for fd,ii in [(fdist_[i],i) for i in fdist_.keys()[:14]]: print fd, ii
```

```
retiradas stopwords feita contagem das palavras em 0.29s
1277 é
1256 não
762 ser
717 participação
548 social
526 sociedade
468 à
459 sobre
367 governo
357 são
337 forma
327 políticas
310 públicas
302 brasil
```

In [75]:
```
print(u"são %i palavras em %i palavras diferentes"%(len(palavras__),len(fdist_)))
```

```
são 91361 palavras em 14653 palavras diferentes
```

In [76]:
```
# para radicalizar (lematização é similar)
# NOW=time.time()
#stemmer = k.stem.RSLPStemmer()
#palavras___=[stemmer.stem(pp) for pp in palavras__]
#fdist__=k.FreqDist(palavras___)
#print("feita freq dist (radicalizada) em %.2f"%(time.time()-NOW,))
```

Escolhendo as palavras mais frequentes para fazer caracterização das mensagens:

In [77]:
```
# escolhendo as 200 palavras mais frequentes
palavras_escolhidas=fdist_.keys()[:200]
```

Extraindo atributos (contagem das palavras) e fazendo classificação bayesiana ingênua. Note que os rótulos "pos" e "neg" estão sendo atribuídos ao acaso. Para aproveitamento, é necessário que sejam aproveitados os dados rotulados, provavelmente pelo pessoal da comunicação.

In [78]:
```
def document_features(documento):
    features={}
    for palavra in palavras_escolhidas:
        features["contains(%s)"%(palavra,)]=(palavra in documento)
    return features
msgsP= [(rr["texto"]["value"],"pos") for rr in msgs[:500]]
msgsN=[(rr["texto"]["value"],"neg") for rr in msgs[500:1000]]
msgsT=msgsP+msgsN
random.shuffle(msgsT)
feature_sets=[(document_features(msg[0]),msg[1]) for msg in msgsT]
train_set, test_set = feature_sets[:500], feature_sets[500:]
classifier = k.NaiveBayesClassifier.train(train_set)
```

Mostrando as características mais informativas:

In [79]:
```
classifier.show_most_informative_features(5)
```

```
Most Informative Features
        contains(comitê) = True              pos : neg    =      4.1 : 1.0
          contains(hoje) = True              pos : neg    =      4.0 : 1.0
         contains(saúde) = True              neg : pos    =      3.1 : 1.0
        contains(sugiro) = True              neg : pos    =      2.7 : 1.0
         contains(grupo) = True              pos : neg    =      2.4 : 1.0
```

Precisão nos dados de teste dummy ≈ 0.5 pois os rótulos não foram atribuídos com dados reais.

In [80]:
```
k.classify.accuracy(classifier, test_set)
```

Out[80]:
```
0.526
```

Classificação de um documento:

In [81]:
```
classifier.classify(document_features(msgsT[12][0]))
```

Out[81]:
```
'neg'
```

|||--- FIM ---|||

2. Seleção de conteúdo por rankeamento (e.g. por tamanho de palavras) e limiar (e.g. número mínimo de palavras)

importanto bibliotecas úteis:

In [54]:
```
from SPARQLWrapper import SPARQLWrapper, JSON
import time, numpy as n, nltk as k
```

Definição de prefixos úteis para as triplas rdf:

In [55]:
```
PREFIX="""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ops: <http://purl.org/socialparticipation/ops#>
PREFIX opa: <http://purl.org/socialparticipation/opa#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX tsioc: <http://rdfs.org/sioc/types#>
PREFIX schema: <http://schema.org/>"""
```

Buscando todos os comentários no endpoint SparQL disparado pelo Jena:

In [56]:
```
NOW=time.time()
q="SELECT ?comentario ?titulo ?texto WHERE \
           {?comentario dc:type tsioc:Comment.\
          OPTIONAL {?comentario dc:title ?titulo . }\
           OPTIONAL {?comentario schema:text ?texto .}}"
sparql3 = SPARQLWrapper("http://localhost:82/participabr/query")
sparql3.setQuery(PREFIX+q)
sparql3.setReturnFormat(JSON)
results4 = sparql3.query().convert()
print("%.2f segundos para puxar todos os comentários do Participa.br"%(time.time()-NOW,))
```

```
2.34 segundos para puxar todos os comentários do Participa.br
```

Limpando mensagens das sugeiras do BD:

In [57]:
```
msgs_=results4["results"]["bindings"]
msgs=[mm for mm in msgs_ if ("titulo" not in mm.keys()) or (("teste de stress" not in mm["titulo"]["value"].lower())
      and ("comunidade de desenvolvedores e nesse caso, quanto mais" not in mm["texto"]["value"].lower()))]
```

Fazendo função para extrair atributos das mensagens:

In [58]:
```
import string, numpy as n
exclude = set(string.punctuation+u'\u201c'+u'\u2018'+u'\u201d'+u'\u2022'+u'\u2013')
def atributos(__msg):
    texto=__msg["texto"]["value"]
    texto_ = ''.join(ch for ch in texto if ch not in exclude)
    palavras=texto_.split()
    tams=[]
    for palavra in palavras:
        tams.append(len(palavra))
    return len(tams), n.mean(tams), n.std(tams)
```

Criando vetor de atributos de cada mensagem:

In [59]:
```
atrs=[]
for msg in msgs:
    atrs.append(atributos(msg))
atrs_=n.array(atrs)
```

Fazendo seleção das mensagens que possuem entre 110 e 115 mensagens:

In [60]:
```
max_palavras=115
min_palavras=110
n_msgs=((atrs_[:,0]>min_palavras)*(atrs_[:,0]<max_palavras)).sum()
print(u"são %i mensagens com mais de %i palavras e menos de %i"%
      (n_msgs, min_palavras, max_palavras) )
```

```
são 26 mensagens com mais de 110 palavras e menos de 115
```

Selecionando mensagems com média do tamanho da palavra maior que 6 caractéres:

In [61]:
```
msgs_i=((atrs_[:,0]>min_palavras)*(atrs_[:,0]<max_palavras)).nonzero()[0]
textos=[[msgs[i]["texto"]["value"] for i in msgs_i][j]
```

```
        for j in (atrs_[msgs_i][:,1]>6.).nonzero()[0]]
for texto in textos: print texto+"\n======="
```

```
1. Auxiliar na divulgação das informações jurídicas para estrangeiros por meio de projetos de longo prazo dentro no âmbito
das próprias universidades (ex. immigration/refugee law clinics), estimulando-se a parceria com as instituições,
organizações e programas/projetos relacionados à temática migratória;
2. Inserir na nova política migratória a capacitação/formação específica para agentes públicos (obrigatória) que atuam ou
possam vir a atuar em razão de suas atribuições institucionais com migrantes, refugiados e apátridas. Nesse sentido, a
experiência e contribuição da academia será fundamental do ponto de vista da expertise científica (ensino e pesquisa -
colaboração para o conteúdo programático da capacitação), além de reforçar a importância do ensino do Direito Internacional
e Direitos Humanos nas Universidades.

=======
ROSAS, Agostinho da Silva; MELO NETO, José Francisco de. Educação Popular: Enunciados Teóricos. João Pessoa: Editora
Universitária da UFPB, 2008.

Os autores buscam delimitar um campo da educação em que seja possível delinearem-se características que apenas à educação
popular façam parte. Trata-se de um esforço de apresentação de constituintes com dimensões formadoras, tendo como aspectos
centrais as experiências de vários educadores populares e as reflexões em desenvolvimento na época. Nessa perspectiva, são
abordadas a experiência histórica, a cultura, o popular, a realidade, o trabalho, a autonomia, a liberdade e a igualdade
como componentes fundantes para a realização de práticas em educação popular, lastreados pela dimensão ética do diálogo.

Disponível em:
http://www.prac.ufpb.br/copac/extelar/producao_academica/livros/pa_l_2004_educacao_popular_enunciados_teoricos.pdf
=======
```

Selecionando mensagems com média do tamanho da palavra menos que 5 caractéres:

In [62]:
```
textos=[[msgs[i]["texto"]["value"] for i in msgs_i][j]
        for j in (atrs_[msgs_i][:,1]<5.).nonzero()[0]]
for texto in textos: print texto+"\n======="
```

```
Seria suave aos ouvidos ver que a população participando e contribuindo pode notar que seus desejos desde que na medida do
possível estão sendo ouvidos e colocado em prática.
A população cansou, pois até coloca sua cara pra bater, porém na hora de colocar em prática o que tanto desejam nada
acontece.
O que resta, a saber, se quando a população participar da construção das políticas publicas, deixara suas vontades o campo
da teoria para o campo da prática, muito difícil isso ocorrer ou talvez impossível, pois o desejo da população praticamente
em todos as vezes e momentos que é aferida suas vontades, são deixados de lado, isso é uma pena.

=======
O §1º é a atividade ou ação natural, o principio de oficio ou por assim dizer a o fato singular do GT.

O §2º é a preposições facultadas que podem ou não advir, em conformidade das necessidades observadas na decorrência do §1º,
ou seja, dependente do §1º.

O §3º é mais fácil, pois não há registros do que não aconteceu, então a referencia, por lógica, deve ser a ultima.

Não me ative em pensar em mesmo método com os incisos, por serem muitos e não existe espaço para estabelecer a desfiguração,
observando que o contexto das determinações estarem em elevação de importância, mesmo que a ordenação ou 'cronologia' traga
maior facilidade de compreensão.
=======
Laura, essa alteração legislativa é muito importante. É muito desproporcional exigir que o migrante tenha que indicar uma
repartição consular em outro país para obter o visto. O fundamento está no fato de que o Estatuto do Estrangeiro não permite
concessão de visto a quem está irregular. Mas é possível uma interpretação sistemática e razoável da norma, para que seja
feita uma distinção para os estrangeiros indocumentados (que estão ainda sem o visto mas que têm o direito à regularização
migratória, como por exemplo por motivo de reunião familiar). Exigir que o estrangeiro que já está no Brasil tenha que sair
do país apenas para buscar o visto é muito oneroso e desproporcional.
=======
```

Fazendo contagem das palavras mais frequentes para seleção:

In [63]:
```
NOW=time.time()
import string, nltk as k
palavras=string.join([i["texto"]["value"].lower() for i in msgs])
exclude = set(string.punctuation+u'\u201c'+u'\u2018'+u'\u201d'+u'\u2022'+u'\u2013')
palavras = ''.join(ch for ch in palavras if ch not in exclude)
palavras_=palavras.split()
print(u"feita lista de todas as palavras de todos os comentários em %.2f"%(time.time()-NOW,))
stopwords = set(k.corpus.stopwords.words('portuguese'))
palavras__=[pp for pp in palavras_ if pp not in stopwords]
fdist_=k.FreqDist(palavras__)
print("retiradas stopwords feita contagem das palavras em %.2f"%(time.time()-NOW,))
```

```
feita lista de todas as palavras de todos os comentários em 0.17
retiradas stopwords feita contagem das palavras em 0.45
```

Fazendo seleção das 14 palavras mais incidentes nos comentários do Participa.br:

In [64]:
```
for fd,ii in [(fdist_[i],i) for i in fdist_.keys()[:14]]: print fd, ii
```

```
1277 é
1256 não
762 ser
717 participação
548 social
526 sociedade
```

```
468 à
459 sobre
367 governo
357 são
337 forma
327 políticas
310 públicas
302 brasil
```

3. Síntese das redes de amizade e de interação do Participa.br

Importando bibliotecas:

In [89]:
```
from SPARQLWrapper import SPARQLWrapper, JSON
import time, numpy as n, networkx as x
```

Definição de prefixos úteis para as buscar dados em contexto semântico:

In [90]:
```
PREFIX="""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ops: <http://purl.org/socialparticipation/ops#>
PREFIX opa: <http://purl.org/socialparticipation/opa#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX tsioc: <http://rdfs.org/sioc/types#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX schema: <http://schema.org/>"""
```

Buscando amizades do Participa.br:

In [91]:
```
NOW=time.time()
q="""SELECT DISTINCT ?aname ?bname
       WHERE {
          ?a foaf:knows ?b .
          ?a foaf:name ?aname .
          ?b foaf:name ?bname .
       }"""
sparql3 = SPARQLWrapper("http://localhost:82/participabr/query")
sparql3.setQuery(PREFIX+q)
sparql3.setReturnFormat(JSON)
results4 = sparql3.query().convert()
print("%.2f segundos para puxar todas as amizades do Participa.br"%
      (time.time()-NOW,))
```

```
0.07 segundos para puxar todas as amizades do Participa.br
```

Erigindo rede de amizades como um grafo não direcionado:

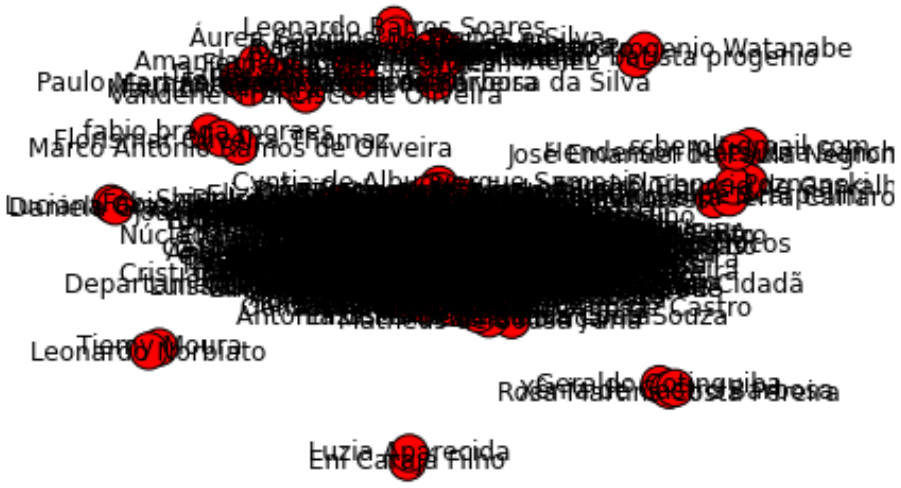
In [92]:
```
g=x.Graph()
for amizade in results4["results"]["bindings"]:
    nome1=amizade["aname"]["value"]
    nome2=amizade["bname"]["value"]
    g.add_edge(nome1,nome2)
```

In [93]:
```
print(u"são %i amizades entre %i pessoas no Participa.br"%
      (g.number_of_edges(), g.number_of_nodes()))
```

```
são 910 amizades entre 443 pessoas no Participa.br
```

Para visualizar, a estrutura já está pronta. Ex:

In [94]:
```
x.draw(g,pos=x.layout.fruchterman_reingold_layout(g))
```



Puxando as interações no Participa.br

In [95]:
```
NOW=time.time()
q2="""SELECT DISTINCT ?aname ?bname
       WHERE {
           ?comentario dc:type tsioc:Comment.
           ?participante1 ops:performsParticipation ?comentario.
           ?participante1 foaf:name ?aname.
           ?artigo sioc:has_reply ?comentario.
           ?participante2 ops:performsParticipation ?artigo.
           ?participante2 foaf:name ?bname.
       }"""
sparql3.setQuery(PREFIX+q2)
sparql3.setReturnFormat(JSON)
results = sparql3.query().convert()
print("%.2f segundos para puxar as interações do Participa.br"%
      (time.time()-NOW,))
```

```
15.58 segundos para puxar as interações do Participa.br
```

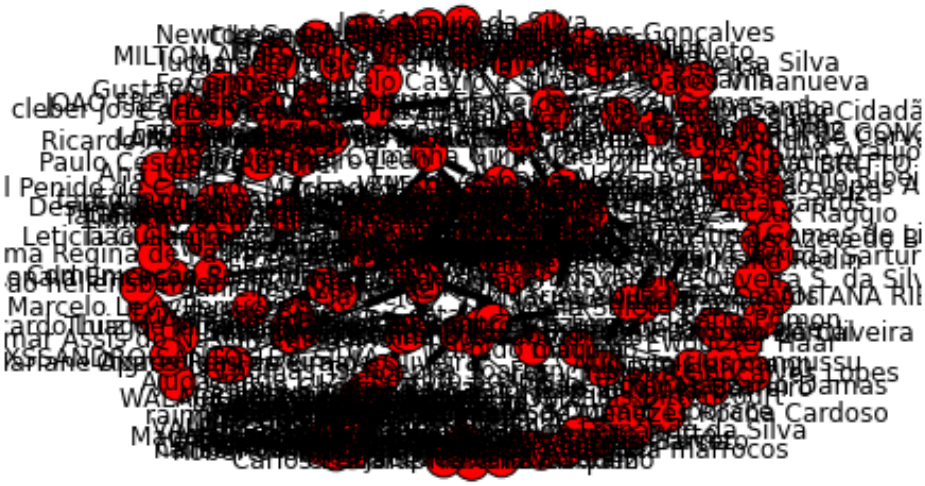
Sintetizando rede direcionada de interação:

In [96]:
```
d=x.DiGraph()
for interacao in results["results"]["bindings"]:
    nome_chegada=interacao["aname"]["value"]
    nome_partida=interacao["bname"]["value"]
    if (nome_partida,nome_chegada) in d.edges():
        d[nome_partida][nome_chegada]["weight"]+=1
    else:
```

```
        d.add_edge(nome_partida,nome_chegada,weight=1.)
```

Estrutura pronta para análises e visualizações:

In [97]:
```
x.draw(d,pos=x.layout.fruchterman_reingold_layout(d))
```



Elencando as 15 pessoas mais conectadas via **interação**:

In [98]:
```
import operator
sorted_d = sorted(d.degree().iteritems(), key=operator.itemgetter(1))
sorted_d[::-1][:15]
```

Out[98]:
```
[(u'Portal', 46),
 (u'Renato Fabbri', 26),
 (u'Gabriela Valle', 19),
 (u'Hylton Sarcinelli Luz', 15),
 (u'F\xe1dia Rebou\xe7as', 13),
 (u'Grazielle Machado', 12),
 (u'Jose mendon\xe7a Furtado Neto', 11),
 (u'andre luiz da silva', 11),
 (u'Daniel Pitangueira de Avelino', 10),
 (u'Henrique Parra Parra Filho', 9),
 (u'Juliano Geraldi', 8),
 (u'Roberto Kodama', 7),
 (u'Marcelo Rodrigues Saldanha da Silva', 7),
 (u'L\xedgia Maria Alves Pereira', 7),
 (u'Frank Lane', 7)]
```

Elencando as 15 pessoas mais conectadas via **amizades**:

In [99]:
```
sorted_g = sorted(g.degree().iteritems(), key=operator.itemgetter(1))
sorted_g[::-1][:15]
```

Out[99]:
```
[(u'Renato Fabbri', 133),
```

Out[99]:
```
[(u'Marcelo Branco', 122),
 (u'Maria Jos\xe9lia Amaral de Menezes', 60),
 (u'Ana C\xe9lia Costa', 36),
 (u'Vicente Aguiar', 35),
 (u'Laura Zacher', 33),
 (u'Luis Felipe Coimbra Costa', 32),
 (u'Ricardo Poppi', 31),
 (u'LUCAS MOREIRA DE SOUSA', 30),
 (u'Ronald Emerson Scherolt da Costa', 29),
 (u'Valessio Brito', 27),
 (u'Grazielle Machado', 20),
 (u'Andr\xe9 Filipe de Assun\xe7\xe3o e Brito', 20),
 (u'Renato Fabbri', 20),
 (u'Daniela Feitosa', 18),
 (u'Daniel Pitangueira de Avelino', 18)]
```

4. Classificação de conteúdo via conectividade dos participantes

Importadas as bibliotecas:

In [10]:
```
from SPARQLWrapper import SPARQLWrapper, JSON
import time, operator, numpy as n, networkx as x
```

Guardados os prefixos mais úteis no domínio:

In [31]:
```
PREFIX="""PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ops: <http://purl.org/socialparticipation/ops#>
PREFIX opa: <http://purl.org/socialparticipation/opa#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX tsioc: <http://rdfs.org/sioc/types#>
PREFIX sioc: <http://rdfs.org/sioc/ns#>
PREFIX schema: <http://schema.org/>"""
```

Baixadas todas as amizades do participa

In [51]:
```
NOW=time.time()
q="""SELECT DISTINCT ?a ?b ?aname ?bname
       WHERE {
          ?a foaf:knows ?b .
          ?a foaf:name ?aname .
          ?b foaf:name ?bname .
       }"""
sparql3 = SPARQLWrapper("http://localhost:82/participabr/query")
sparql3.setQuery(PREFIX+q)
sparql3.setReturnFormat(JSON)
results4 = sparql3.query().convert()
print("%.2f segundos para puxar todas as amizades do Participa.br"%
      (time.time()-NOW,))
```

```
0.12 segundos para puxar todas as amizades do Participa.br
```

Feita rede de amizades do Participa.br:

In [52]:
```
g=x.Graph()
for amizade in results4["results"]["bindings"]:
    nome1=amizade["a"]["value"]
    nome2=amizade["b"]["value"]
    g.add_edge(nome1,nome2)
```

Baixadas interações no Participa.br:

In [17]:
```
NOW=time.time()
q2="""SELECT ?participante1 ?participante2 ?aname ?bname
       WHERE {
           ?comentario dc:type tsioc:Comment.
           ?participante1 ops:performsParticipation ?comentario.
           ?participante1 foaf:name ?aname.
           ?artigo sioc:has_reply ?comentario.
           ?participante2 ops:performsParticipation ?artigo.
           ?participante2 foaf:name ?bname.
       }"""
sparql3.setQuery(PREFIX+q2)
sparql3.setReturnFormat(JSON)
results = sparql3.query().convert()
print("%.2f segundos para puxar as interações do Participa.br"%
      (time.time()-NOW,))
```

```
16.06 segundos para puxar as interações do Participa.br
```

Feita rede de interações:

In [18]:
```
d=x.DiGraph()
for interacao in results["results"]["bindings"]:
    nome_chegada=interacao["participante1"]["value"]
    nome_partida=interacao["participante2"]["value"]
    if (nome_partida,nome_chegada) in d.edges():
        d[nome_partida][nome_chegada]["weight"]+=1
    else:
        d.add_edge(nome_partida,nome_chegada,weight=1.)
```

Listando os mais conectados para ver suas URIs e verificar bom andamento das buscas SparQL:

In [53]:

In [53]:
```
sorted_g = sorted(g.degree().iteritems(), key=operator.itemgetter(1))
sorted_g[::-1][:15]
```

Out[53]:
```
[(u'http://participa.br/profile/marcelobranco', 122),
 (u'http://participa.br/profile/mjade', 60),
 (u'http://participa.br/profile/anita', 36),
 (u'http://participa.br/profile/vicentedeaguiar', 35),
 (u'http://participa.br/profile/laurazacher', 33),
 (u'http://participa.br/profile/lfelipe', 32),
 (u'http://participa.br/profile/ricardopoppi', 31),
 (u'http://participa.br/profile/lucasmoreira', 30),
 (u'http://participa.br/profile/ronald.costa', 29),
 (u'http://participa.br/profile/valessiobrito', 27),
 (u'http://participa.br/profile/grazi_machado', 20),
 (u'http://participa.br/profile/decko', 20),
 (u'http://participa.br/profile/rfabbri', 20),
 (u'http://participa.br/profile/nilzetepacheco', 18),
 (u'http://participa.br/profile/cclaro', 18)]
```

In [54]:
```
sorted_d = sorted(d.degree().iteritems(), key=operator.itemgetter(1))
sorted_d[::-1][:15]
```

Out[54]:
```
[(u'http://participa.br/profile/portal', 46),
 (u'http://participa.br/profile/rfabbri', 26),
 (u'http://participa.br/profile/gabriela', 19),
 (u'http://participa.br/profile/hyltonsarcinelliluz', 15),
 (u'http://participa.br/profile/fadia', 13),
 (u'http://participa.br/profile/grazi_machado', 12),
 (u'http://participa.br/profile/josefurtado', 11),
 (u'http://participa.br/profile/andre61', 11),
 (u'http://participa.br/profile/davelino', 10),
 (u'http://participa.br/profile/parrahenri', 9),
 (u'http://participa.br/profile/julianogeraldi', 8),
 (u'http://participa.br/profile/capuano', 7),
 (u'http://participa.br/profile/kodama', 7),
 (u'http://participa.br/profile/ibebrasil', 7),
 (u'http://participa.br/profile/thiagozoroastro', 7)]
```

Pode-se selecionar mensagens de periféricos, hubs e intermediários. Por simplicidade, aqui estão 5 mensagens dos 5 mais conectados nas **atividades**:

In [55]:
```
uris=[i[0] for i in sorted_d[::-1][:5]]
for uri in uris:
    q=""" SELECT ?texto
       WHERE {
          ?comentario dc:creator <%s> .
          ?comentario dc:type tsioc:Comment.
          ?comentario schema:text ?texto .
       } LIMIT 2"""%(uri,)
    #print q
    sparql3.setQuery(PREFIX+q)
    results4 = sparql3.query().convert()
    #print results4
    print "\n"
    try:
        print uri+"\n"+ results4["results"]["bindings"][0]["texto"]["value"]
    except:
        pass
```

```
http://participa.br/profile/rfabbri
que órgãos, de que programas?


http://participa.br/profile/gabriela
Não pode esquecer da gestão democrática nas universidades e instituições de ensino.
Vale a pena mencionnar isso espqecificamente, porque já tem lei prevendo, mas na prática não acontece.


http://participa.br/profile/hyltonsarcinelliluz
Apoio a sugestão de incluir o termo &amp;quot;Controle&amp;quot; ao nome da politica em pauta, com vista a uniformizar a conceituação, visto que já existe e é difundido o termo &amp;quot;controle social&amp;quot; para políticas públicas, no qual se insere a participação em debate.



http://participa.br/profile/fadia
O termo participação já está em ampla utilização no âmbito dos governos, apropriado inclusive pelo capital. O que se questiona é o que se entende por essa participação e o que se quer é a efetivação da participação. Concordo com a inserção do termo controle.

Política Nacional para Efetivação do Controle e Participação Social
```

Pode-se selecionar mensagens de periféricos, hubs e intermediários. Por simplicidade, aqui estão 5 mensagens dos 5 mais conectados nas **amizades**:

In [56]:
```
uris=[i[0] for i in sorted_g[::-1][:5]]
```

```
for uri in uris:
    q=""" SELECT ?texto
       WHERE {
          ?comentario dc:creator <%s> .
          ?comentario dc:type tsioc:Comment.
          ?comentario schema:text ?texto .
       } LIMIT 2"""%(uri,)
    #print q
    sparql3.setQuery(PREFIX+q)
    results4 = sparql3.query().convert()
    #print results4
    print "\n"
    try:
        print uri+"\n"+ results4["results"]["bindings"][0]["texto"]["value"]
    except:
        pass
```

```
http://participa.br/profile/marcelobranco
Uma foto ou figura em cada post ficaria muito melhor.



http://participa.br/profile/anita
Oiiii Ronald!!!


http://participa.br/profile/vicentedeaguiar
Por exemplo, essa sua resposta que você fez no meu comentário, eu não consigo visualizar no mural, só diretamente no artigo. Vo
consegue ver no mural essa minha resposta agora? Se sim, pode ser problema na versão do CSS para Firefox 25.0.1.


http://participa.br/profile/laurazacher
Sejam muito bem-vindos!

Iniciamos a Conferência Virtual da DPU sobre Migrações e Refúgio já com uma importante discussão!

Aproveito o início da discussão reiterando a manifestação da Dra. Ana Luisa e repassando algumas orientações para organizarmos
processo de discussão coletiva:

- Inicialmente, a discussão será em sentido amplo; contudo, conforme a discussão avançar, é preciso que os tópicos postados ind
(1) a qual dos eixos temáticos estão vinculados, (2) a qual bloco está contemplando; (3) qual temática aborda.

Exemplo: Discussão proposta sobre expulsão: inlcuir no título da mensagem:

Eixo Temático nº (I ou IV) - Bloco: Mudanças Legislativas - Tema: Expulsão

Relembro os eixos temáticos a serem discutidos na Conferência da DPU:

Eixo I - Igualdade de tratamento e acesso a serviços e direitos;

Eixo IV - Abordagem de violações de direitos e meios de prevenção e proteção, no qual também será discutido o Subtema -
Enfrentamento ao Sequestro/ Subtração Internacional de Crianças e
a aplicação da Convenção de Haia de 1980 sobre os Aspectos Civis do Sequestro Internacional de Crianças.

Em cada uma das mensagens postadas, favor indicar a qual dos blocos propostos está vinculada:

(a) mudanças legislativas (lei em sentido amplo);
(b) políticas públicas;
(c) atuação e estruturação da DPU para assistência ao migrante.

- Estas orientações visam facilitar o processo de sistematização das propostas. Para ilustrar o produto final da sistematização
deixo o link para o Caderno de Propostas da 1ª Conferência Nacional de Segurança Pública, cujo modelo de conferência é a base d
atual Comigrar:

http://www.ipea.gov.br/participacao/images/pdfs/conferencias/Seguranca_Publica/caderno_propostas_1_conferencia_seguranca_public

- Informações sobre os eixos temáticos e sobre o processo da Conferência podem ser obtidas através do Manual do Participante:
http://www.dpu.gov.br/images/stories/arquivos/PDF/Manual_do_Participante_-_DPU_-_COMIGRAR.pdf

- Qualquer dúvida sobre o processo da Conferência, favor entrar em contato através do telefone (51) 3216-6961 e/ou do correio
eletrônico comigrar@dpu.gov.br.

Vamos construir uma política migratória com Justiça e Igualdade para todos!

|||--- FIM ---|||
```